SCTE STANDARDS JOURNAL

VOLUME 2 NO. 3 OCTOBER 2022

SCTE TECHNICAL JOURNAL

SCTE
a subsidiary of CableLabs®

SCTE TECHNICAL JOURNAL

VOLUME 2, NUMBER 3

October 2022

Society of Cable Telecommunications Engineers, Inc.
140 Philips Road, Exton, PA 19341-1318

Table of Contents

4 **Foreword**

5 **Estimating the Energy Usage of Streaming Delivery of Pay-Tv Video Services**
Chuck Carroll, Saras Partners
Rene Spee, Saras Partners
Lew Rakowsky, Saras Partners

31 **Aluminum Cable Applications with Energy Storage in Cable Head Ends and Data Centers**
Carly Waschka, Priority Wire & Cable
Larry Hamilton, Priority Wire & Cable

40 **L4S Transport over DOCSIS: Experiments and Observations of a Low Latency Transport Protocol over DOCSIS Networks**
Ram Ranganathan, CommScope

61 **Optimizing Wi-Fi Channel Selection in a Dense Neighborhood**
Yonatan Vaizman, Comcast
Hongcheng Wang , Comcast

73 **MLOps and ML Platforms: An Overview**
Nicholas Pinckernell, Comcast
Jan Neumann, Comcast

93 **An Emerging Alternative for Meeting Zero-Emissions Goals**
Seth Terry, New Day Hydrogen

***SCTE Engineering Committee Chair and Data Standards Subcommittee (DSS) Chair**:*
David Fellows,
SCTE Member

Senior Editor
Paul Schneider
Paul Schneider Public Relations

Publications Staff
Chris Bastian
SVP & Chief Technology Officer, SCTE

Dean Stoneback
Senior Director, Engineering & Standards, SCTE

Kim Cooney
Document Editor, SCTE

Editorial Correspondence: If there are errors or omissions to the information provided in this journal, corrections may be sent to our editorial department. Address to: SCTE Journals, SCTE, 140 Philips Road, Exton, PA 19341-1318 or email journals@scte.org.

Submissions: If you have ideas or topics for future journal articles, please let us know. All submissions will be peer reviewed and published at the discretion of SCTE. Electronic submissions are preferred, and should be submitted to SCTE Journals ,journals@scte.org.

You may not all share my enthusiasm for the Phillies' run in the Major League Baseball postseason and the Eagles' hot start in the NFL, but let's agree that September was huge for SCTE and the cable broadband technology community.

At IBC, our Event Scheduling and Notification Interface standard captured first place in the CSI Innovation Awards for its ability to support the alternate feeds that are so important to accommodating sports rights restrictions as IP video delivery replaces QAM. Less than two weeks later, SCTE Cable-Tec Expo – the largest cable industry event in the Americas – returned with a roar to a live venue after two years of virtual events, drawing more attendees than it did pre-pandemic and unveiling massive accomplishments and commitments in 10G, sustainability, and more.

This latest edition of the SCTE Technical Journal shows how cable is continuing to power the future of telecommunications. Our authors address multiple topics in the areas of energy and sustainability, broadband, and machine learning. Here's what you'll find:

- "Estimating the Energy Usage of Streaming Delivery of Pay-TV Video Services," by Saras Partners' Chuck Carroll, Rene Spee and Lew Rakowsky.
- "Aluminum Cable Applications with Energy Storage in Cable Head Ends and Data Centers," by Priority Wire & Cable's Carly Waschka and Larry Hamilton.
- "An Emerging Alternative for Meeting Zero-Emissions Goals," by New Day Hydrogen's Seth Terry.
- "L4S Transport over DOCSIS: Experiments and Observations of a Low Latency Transport Protocol over DOCSIS Networks," by CommScope's Ram Ranganathan.
- "Optimizing Wi-Fi Channel Selection in a Dense Neighborhood," by Comcast's Yonatan Vaizman and Hongcheng Wang.
- "MLOps and ML Platforms: An Overview," by Comcast's Nicholas Pinckernell and Jan Neumann.

How the Phillies' and Eagles' seasons wind up will be anybody's guess, but a sure bet is that SCTE thought leadership will continue to advance cable broadband's competitive position and drive the development of new services and opportunities for the industry. I hope you will take a few moments to read these latest technical papers and to put SCTE Cable-Tec Expo 2023 – October 16-19 in Denver – on your calendars.

Thank you for your involvement in SCTE and best wishes for a great Fall and a splendid holiday season.

SCTE STANDARDS JOURNAL

Estimating the Energy Usage of Streaming Delivery of Pay-TV Video Services

A Report Developed for the U.S. Set-Top Box Voluntary Agreement Steering Committee

Chuck Carroll – Saras Partners
Rene Spee – Saras Partners
Lew Rakowsky – Saras Partners

Table of Contents

Title	Page Number
Table of Contents	6
1. Executive Summary	8
2. Introduction	8
3. Video Delivery Platforms & Process Flows	10
3.1. Video Delivery Platforms	12
3.1.1. Legacy Linear TV Delivery by STB	13
3.1.2. Legacy Linear Content Record and Playback via DVR	15
3.1.3. Linear TV Delivery via Video Streaming	15
3.1.4. Linear Content Record via Cloud DVR	18
4. Platform Comparison for Data Analysis	19
4.1. Legacy vs. Streaming Service Delivery Comparison Structure	20
5. Data Used in Study	22
5.1. Service Provider Data	23
5.2. Publicly Sourced Data	23
5.3. Customer Premises Equipment for Legacy Use Cases	23
5.4. Data Compilation	24
6. Summary of Results	25
7. Observations and Future Improvement	26
7.1. Facility Efficiency	27
7.2. cDVR Implementation	27
7.3. Energy Impact of Data Center Component Evolution	27
7.4. Energy Impact of STB Evolution	28
8. Summary and Conclusions	28
9. Abbreviations	29
10. References	29

List of Figures

Title	Page Number
Figure 1 - Linear Television Viewing Service Process	10
Figure 2 - Recording Service Process	11
Figure 3 - Recording Service Playback Process	11
Figure 4 - Generic Video Delivery Platform	12
Figure 5 - Legacy Linear TV Video Platform	13
Figure 6 - Legacy Linear TV Viewing Process Flow	14
Figure 7 - Legacy Record and Playback via DVR Process Flow	15
Figure 8 - Video Streaming Platform	16
Figure 9 - Video Streaming Linear TV Process Flow	17
Figure 10 - cDVR Platform	18
Figure 11 - cDVR Record, Store, and Playback Process Flow	19

Figure 12 - Linear TV Service Delivery Platform Comparison 20

List of Tables

Title **Page Number**

Table 1 - Component Comparison Summary 22

Table 2 - Fielded Non-DVR Set-Top Boxes, Typical Energy Consumption and Weighted Average [4] 24

Table 3 - Fielded DVR, Typical Energy Consumption and Weighted Average [4] 24

Table 4 - Incremental Streaming and cDVR Energy Requirements (kWh/year) per customer 25

Table 5 - Incremental Streaming and cDVR Energy Ranges vs. Legacy STB Viewing (kWh/year) 25

Table 6 - Energy Savings — Streaming and cDVR vs. Replacing Legacy STB (kWh/year) 26

Acknowledgements

We are grateful to the following who contributed their time, expertise and valuable insights to this study, and supported the review process of various drafts of the paper.

- Arman Shehabi – Lawrence Berkeley National Laboratory (LBNL)
- Eric Masanet – University of California, Santa Barbara (UCSB)

1. Executive Summary

This paper compares the energy usage associated with watching streaming linear or recorded video in a typical US home to watching the same content delivered via a set-top box (STB) or digital video recorder (DVR). The analysis does not estimate total energy usage, but rather focuses on usage in hardware and software elements in the streaming/cloud-based infrastructure within the service providers' systems that are incrementally different to the legacy STB and DVR delivery methods. The analysis is also incremental to the separate purchase of Internet access service, which is typical of households that elect streaming options to watch video service.

The four largest pay-TV video service providers in the United States, as signatories to the "Voluntary Agreement for the Ongoing Improvement to the Energy Efficiency of Set-Top Boxes," provided energy data used in this study. To provide for confidentiality to the service providers offering data, the paper uses and reports only averages of variability across responses rather than focusing on individual responses. For STB and DVR energy characteristics, we rely on the annual reporting of energy data by the signatories to the Voluntary Agreement.

The analysis shows that the incremental energy in the network for streaming and cloud-based linear and recorded video services delivered to customers is substantially less than traditional set-top box energy consumption in the home: an average of 3.6 kWh/year for linear TV and 14.4 kWh/year for recorded video. By comparison, the average non-DVR STB fielded between 2014 and 2020 draws 84.6 kWh per year with a range of 49 to 103.3 kWh. For DVRs fielded between 2015 and 2020, the average energy requirement is 151.3 kWh per year, with a range of 134.4 to 170.6 kWh. STB and DVR energy efficiency improvements and migration strategies may narrow the difference over time. However, based on the data provided, we identify some possible avenues for even more improvements in the energy efficiency of streaming/cloud-based delivery.

2. Introduction

In 2012, the first Set-Top Box Voluntary Agreement was adopted, covering the set-top boxes used by approximately 90% of pay-TV customers in the United States. The agreement was expanded and renewed several times, and later a second agreement was adopted by many of the same parties to cover small network equipment for the broadband market. The Set-Top Box Voluntary Agreement establishes commitments by the signatories to continue improvements in the energy efficiency of set-top boxes. Compliance and progress are documented each year in an annual report published by an independent party. Copies of the agreements and annual reports are available at www.energy-efficiency.us.

In this paper, the term "Digital Video Recorder" (DVR) refers to a recording enabled set-top box, while "Set-Top Box" (STB) denotes a traditional, non-DVR piece of equipment.

While most TV viewing and video recording is still achieved using the legacy delivery method with STBs and DVRs located in the customers' premises, the industry has noted that consumers increasingly use internet connected television sets and apps to watch video programming. Thus, it is of interest to compare the energy balance between the legacy delivery method and the internet based streaming equivalent. This report presents an investigation of the difference in energy usage between the two delivery methods. Specifically, we investigate two comparison scenarios:

1. Comparison of energy required to deliver linear TV video content via legacy delivery platforms to television sets using STBs in the home versus providing service via video streaming over the Internet through the service providers' apps.
2. Comparison of energy required to deliver time-shifted (recorded) linear TV content to a television set using in-home DVRs versus providing the capability via cloud-based DVR (cDVR) applications.

For the purpose of this study, the term "linear" refers to the combination of broadcast terrestrial, cable, and satellite content delivered to the home.

Recent projects have addressed the total assessment of energy usage and carbon emissions of the data center, internet transport, and the end-user components associated with viewing of video content for the European market [1,2]. In [1], the overall energy and carbon impact of delivering and viewing British Broadcasting Corporation (BBC) programs over different delivery platforms (terrestrial, satellite, cable, streaming) was analyzed using 2016 data. The study used behavioral models of users' viewing profiles in combination with electricity data to assess energy usage for each method. Although results from the study were not applicable to the comparison being undertaken, the report did find that, unsurprisingly, STBs and DVRs have the largest contribution to the overall energy requirement, accounting for greater than 65% of energy usage of the cable and satellite delivery platforms excluding TV screen energy.

The white paper [2] is also based on European data and specifically focuses on energy usage and carbon impact of video streaming. The study looks at the total energy usage with clearly defined boundaries, considering data centers, transmission, and end-use devices. While not directly applicable to the premise of the current analysis, we note that we were able to corroborate the data center and content delivery network estimate with data obtained from service providers for the current study.

We structured our analysis by initially investigating any differential elements required in the streaming delivery method when compared to the legacy STB and DVR delivery methods as outlined in Sections 3 and 4. Following this, we worked with the four participating service providers to obtain data on energy usage. The responses varied in format and depth, and we homogenized the data points for aggregation and analysis. We used publicly available data to augment the service provider data. To maintain confidentiality we present averages and variability rather than focus on individual responses. Details related to data used in the analysis are provided in Section 5. Section 6 provides a comparison of the energy difference between legacy STB and DVR and streaming delivery methods. The results shown illustrate that streaming and cloud-based video recording are quite efficient when compared to the legacy STB and DVR delivery methods. In Section 7, we discuss some possibilities of further reducing the energy requirements associated with streaming.

As per the scope of this study, our analysis omits network interface devices such as cable modems and optical network terminations (ONTs). This is not unreasonable, given that well over 90% of residential customers [3,9] are receiving broadband service and streaming is incremental to an already existing internet service. Additionally, [1] states that domestic modems and Wi-Fi routers "operate at relatively constant power consumption that vary little with respect to workload", based on typical video streaming traffic. Thus, the network interface is already on the premises for broadband accesss and is not specifically replacing STB or DVR, and the energy usage is not changed as a result of the streaming data being a part of the throughput.

For similar reasons, our analysis omits end use devices customers use to access content in the home. The comparison being made is a "like-for-like" one utilizing the same devices in the home. As STB/DVR primarily connect to television sets, it is assumed streamed content is via streaming apps on the television set itself, excluding use of streaming sticks connected to non-smart television sets. Other end use devices such as smart phones and tablets are not considered in the analysis. As shown in [2], these devices generally use less energy than television sets. As such, excluding them does not create unfair bias, but instead creates potential for improvement as they are incorporated in any future analysis.

3. Video Delivery Platforms & Process Flows

This analysis focuses on presenting representations of video delivery platforms from content ingest to customer viewing for each delivery method, overlaying energy usage on the platform representations, and comparing usage between the alternatives.

The first step is to define the services to be investigated. A clear definition of the service delivered, along with its associated process flow and use case, provides guidance with respect to the hardware, software, cables, etc., needed to deliver the service. It also helps ensure the platform representations used for the comparison are a "like-for-like" comparison between the legacy delivery methods and their streaming/cloud-based alternatives.

The study focuses on the delivery of two services to end subscribers in the home, the linear television viewing service, and time-shifted (recorded) linear content service. The definition of each service follows below. Note in these definitions the term "platform" generically refers to the totality of the components (data center, transport network, and customer premises) along with the associated cables, hardware, and software used to deliver the content, whether it is the legacy or streaming/cloud-based delivery method.

Linear Television Viewing Service: This is defined as a customer viewing linear content on a television set in the home. From a process perspective, a simple generic view of the flow is illustrated in Figure 1.

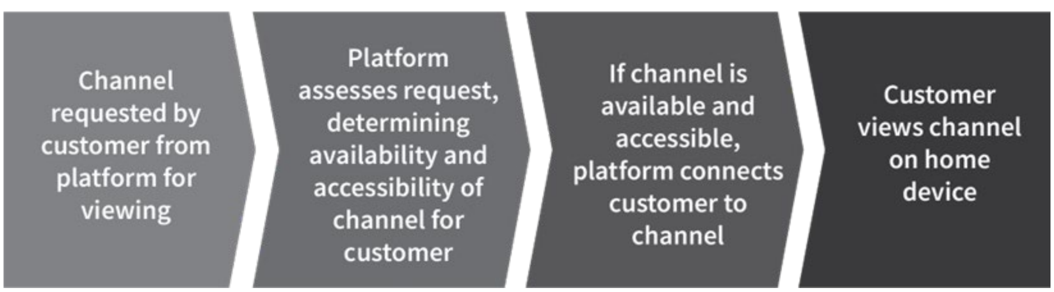


Figure 1 - Linear Television Viewing Service Process

As the flow shows, the process starts with the customer requesting linear content on their television set in the home. The request occurs either via interaction with the STB in the legacy delivery method, or via interaction with an app on the TV in the streaming delivery method. The platform assesses this request, and assuming the content is available, and the customer has access rights, connects the customer to the content (channel) for viewing on the customer TV in the home.

Recorded Linear Content Service: This is defined as a customer recording and storing linear content for viewing on their television set at a later time. As with linear television viewing, the request occurs either via interaction with the DVR in the legacy delivery method, or via interaction with an app on the TV in the streaming delivery method. From a process perspective, this consists of two separate but related processes, the record process, and the playback process. A generic view of the flow for each is shown in Figure 2 and Figure 3.

Figure 2 - Recording Service Process

The record process starts with the customer request to record a linear content asset. The platform assesses the request, determines access and availability similar to the live video viewing process, but instead of delivering to the television set at that time, it records and stores the program for later playback.

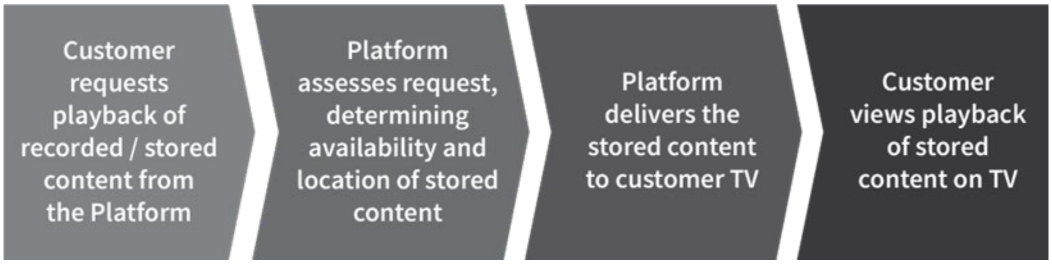


Figure 3 - Recording Service Playback Process

The playback process also starts with a customer request, but in this case to play recorded content. The platform similarly determines content availability and location, ultimately playing it back on the customer TV in the home, assuming it is available for the customer to do so.

It is understood there is much more process-oriented back and forth between customer and platform related to selecting programs and managing the record and playback process. But for the purpose of this analysis, the above use case process flows provide the base ability to develop comparisons.

With the services defined and process flows and use cases understood, platform representations can be created for each of the specific use cases under investigation. To summarize the comparisons to be made:

For linear television viewing:

- Legacy linear TV video delivery via STB to the television set in the home

- Delivery of linear TV via streaming video content to the television set in the home

For recorded viewing of linear television video content:

- Legacy record and playback of linear TV content via in-home DVR on subscriber television set
- Record and playback of linear TV content on subscriber television set via cloud-based DVR

The following sections detail the generic platforms for each of the comparison process flows. We note that the platform representations are not specific to any service provider contributing information but instead reflect a generic view, aggregating functions for service delivery under generic component areas for the analysis.

3.1. Video Delivery Platforms

Utilizing a combination of public data as well as data received from the service providers, a generic representation of a platform for video delivery can be developed. As shown in Figure 4, from a high-level perspective, the video delivery platform has three main components:

Figure 4 - Generic Video Delivery Platform

Data Center Component: Data centers constitute the facilities where the centralized functions associated with video content preparation and delivery to subscribers are performed. Data centers are purpose-built facilities housing the complex high-end server and storage elements used for the service provider's video, voice, and broadband services. Typically, service providers centralize these functions in as few facilities as practicable, with data center elements serving tens of thousands to millions of subscribers depending on service provider size. It should be noted that for our analysis the term data center is generically used to cover all facilities where these functions are performed, the term function refers to a task performed in a component, and the term element is meant to reference the hardware devices and associated software utilized within a function and / or platform component.

Transport Network Component: The transport network is made up of the network cables and equipment used to efficiently transport video content from the data centers to the customer premises. Transport is performed over a variety of facilities including fiber, copper, coax, and satellite, and is shared between video content and the other network services such as internet and voice. As networks typically are distributed in nature, transport network elements can service anywhere from tens of customers in the last mile of the transport network, up to tens or hundreds of thousands of subscribers in the core and backbone portions.

Customer Premises Component: These constitute the devices placed in the home to transition the signals from the transport network to viewable video content for the subscriber, as well as manage the process of connecting to and / or utilizing the service. The term device in this paper is differentiated from the term element in that a device is used to denote equipment specific to a subscriber.

All video delivery platforms analyzed are structured in this manner, with functions, elements, and devices placed in each of these three component areas as appropriate.

3.1.1. Legacy Linear TV Delivery by STB

Linear television delivery to customers historically has been via delivery platforms geared to deliver large amounts of channel capacity to a subscriber home for viewing. Whether the access network is hybrid-fiber coax (HFC), fiber-to-the-home (FTTH), or satellite, it typically broadcasts the same sets of channels to a large number of subscribers. These platforms have a large amount of bandwidth for channel capacity delivered from the platform to the customers, with no or minimal data transported back from the customer for video purposes. The platforms rely on edge devices in the home (STBs) to transition network capacity into a useable service for the customer. The STB provides customer devices with the channel they select from the broad array of channels available.

Figure 5 shows a generic legacy platform for linear video content viewing as used in this comparison study.

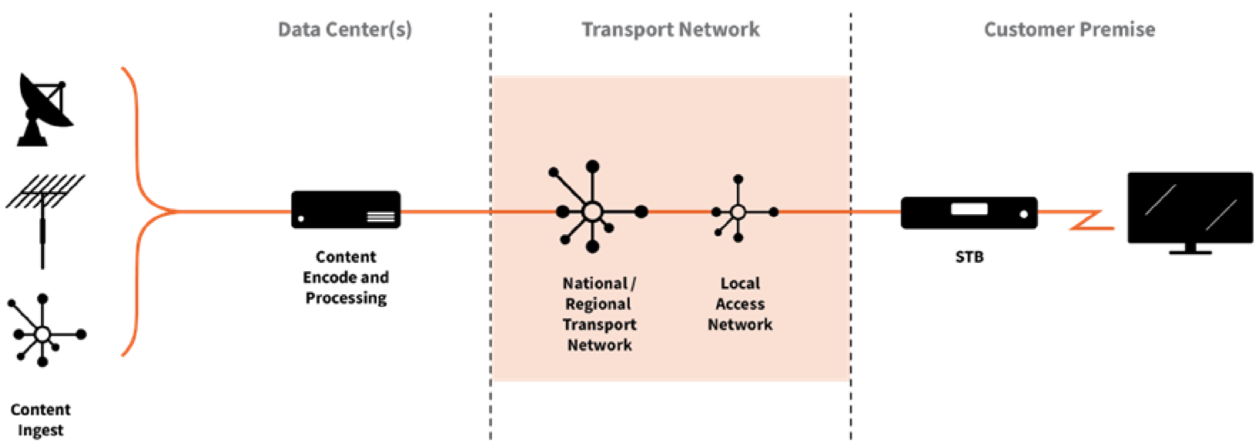


Figure 5 - Legacy Linear TV Video Platform

Key components of the platform include:

- Content Ingest: Content is ingested from satellite feeds, over-the-air feeds, or via the internet. Content is ingested at a number of different points, with national content ingested centrally, and regional/local content ingest occurring in the regional/local area where the content originates.
- Data Center: Ingested content is encoded and processed in facilities for delivery to the transport network. Legacy delivery method data center processing of the content includes grooming and rate-shaping functions meant to prepare the content for efficient delivery of the content's standard definition (SD) / high definition (HD) bitstream across the network to the subscriber. Encode and processing functions are typically performed by a variety of specific hardware elements, placed in some combination of service provider data center, hub, and / or head-end facilities.
- Transport Network: The transport network moves the content bits from service provider facilities to the customer premises. If there are national / regional networks involved in the

transport, the content bitstreams typically are packaged and transported via IP on high-speed fiber networks.

Once the content arrives at the edge facility, the network transitions to the access network for delivery of the content bitstreams to the premises. Delivery in this portion of the network typically consists of some combination of internet protocol (IP) and / or quadrature amplitude modulation (QAM) or quadrature phase shift keying (QPSK) based structures across either fiber, coax, copper, or satellite physical access networks. We would note that mobile networks are excluded from this analysis, as they are typically not used to connect home TV sets with these services.

- Customer Premises Equipment (CPE): At the customer premises, the network is connected to the STB. The STB acts as an edge device, translating customer requests for viewing content into connection to the appropriate broadcast video channel bitstream delivered from the network. The STB includes some nominal processing capability needed for providing a user interface, as well as other functions necessary to deliver customer requested content (such as authentication, authorization, decryption, and decompression).

Overlaying linear TV use case processes yields Figure 6 below.

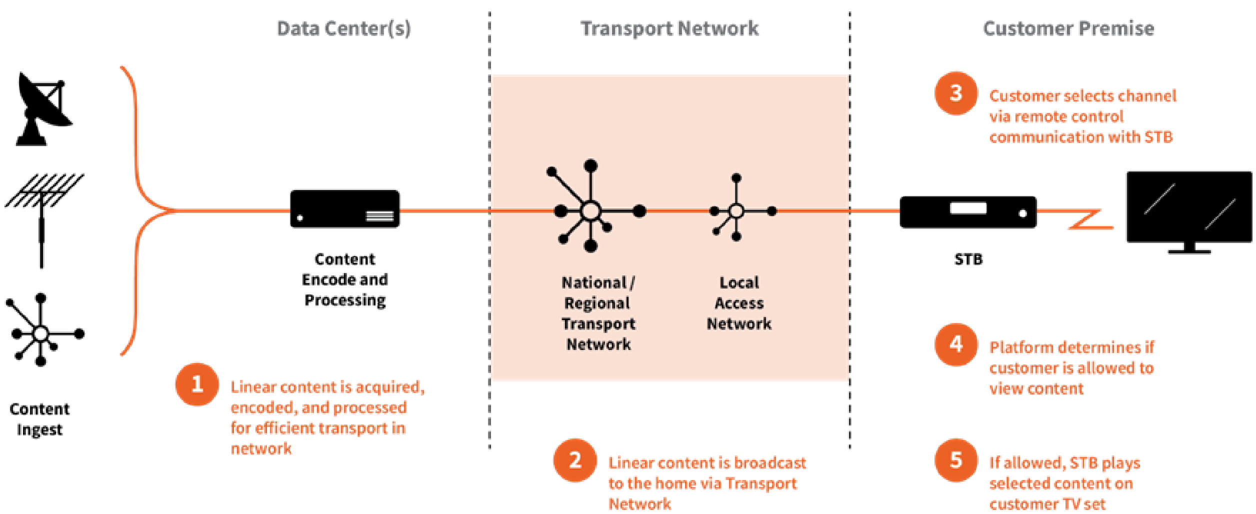


Figure 6 - Legacy Linear TV Viewing Process Flow

From a process perspective, content is ingested and processed at service provider facilities. Processing utilizes the required encoding/transcoding to make transport streams compatible for use in the service provider network. The content is aggregated and delivered across the transport network into the customer premises. In the premises the STB connects to the transport network and the customer end device (television set) and provides an interface with which the customer can interact with the device to select and view content (e.g. remote control). To view content the customer interacts via the STB user interface, selecting content to be viewed. If the customer has permission to view the selected content, the STB connects the customer to the content for viewing. Outside of small interactions related to permissions, the customer channel selection and interaction is managed locally by the STB.

3.1.2. Legacy Linear Content Record and Playback via DVR

The ability to record and store linear content for time-shifted viewing historically has been accomplished via adding necessary technology in the home to implement this application. When television signals were analog, customers purchased devices that recorded programs on physical tapes. With the advent of digital video, service providers added storage to STBs, creating the ability for the STB to physically store customer selected programs for playback later. Software including user interfaces were added to the STB platform for the record and playback, assisting the customer in managing the process.

Consequently, legacy record and playback via DVR builds on the platform shown in Figure 5. The change is at the customer premises, where a different STB is required, inclusive of the hardware/software needed to support the recording and storage of linear content, for playback at a later time. This type of STB is called a DVR in this analysis.

The overlay of the process flow on the platform elements and the service process flow for the record and playback of content is shown in Figure 7

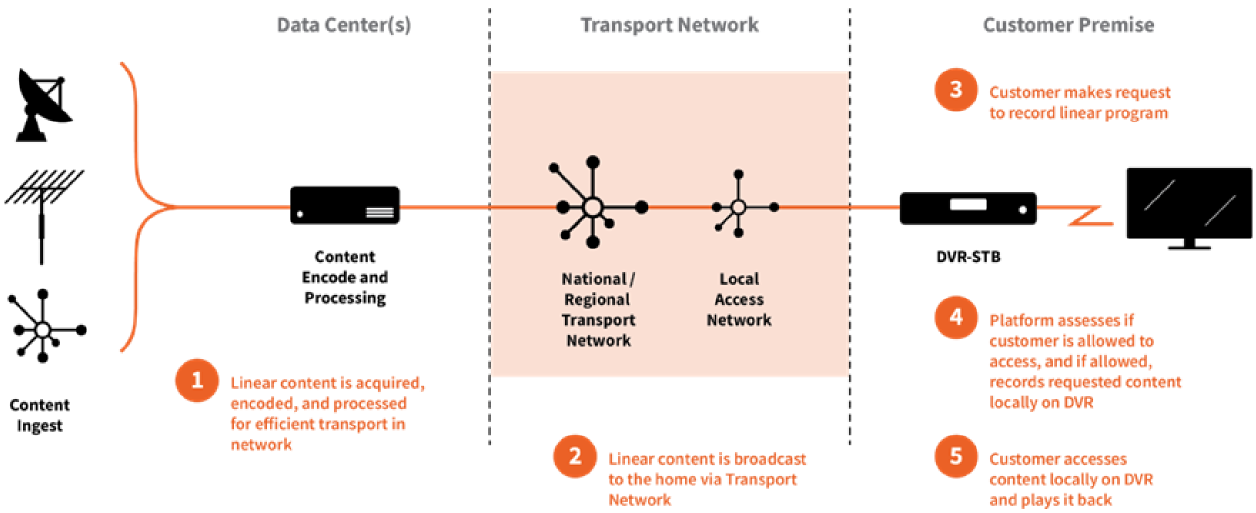


Figure 7 - Legacy Record and Playback via DVR Process Flow

From a process perspective, content is delivered to the home in the same way as it is for linear television viewing via STB. To record a program, the customer interacts via the DVR user interface, selecting content to be recorded. The DVR manages the record process and, if the customer has permission to view the selected content, ensures the content is recorded as requested. Recorded content is stored locally on the DVR memory. When the customer wishes to play back the content, the customer interacts with the DVR user interface to initiate and manage the playback process. As with linear TV viewing via STB, outside of small interactions related to permissions, the customer interactions for this function are managed locally by the DVR.

3.1.3. Linear TV Delivery via Video Streaming

Streaming video is traditionally defined as video delivered to end customers via the Internet. Figure 8 shows a generic network platform for video streaming of linear content used for the comparative analysis.

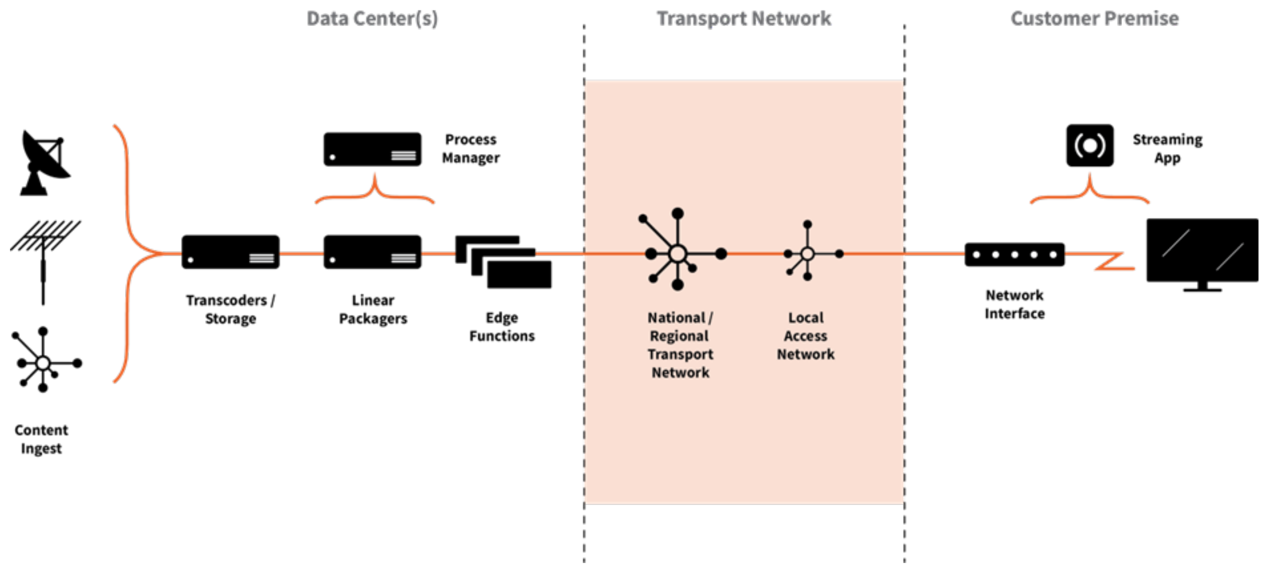


Figure 8 - Video Streaming Platform

Key elements of the delivery platform include:

Data Center: for streaming, data centers provide more complex functionality than in legacy systems by integrating the following functions:

- Content Ingest: Video streaming for linear TV delivery generally uses the same linear TV ingested content as is used for legacy broadcast linear TV delivery.
- Transcoding: Once the content is received in the data center location, it is encoded and/or transcoded into the multiple video format(s) required to service the variety of end user devices in the home. Transcoding for streaming differs from legacy processing in that it includes transcoding and storing content in formats related to playback on the wide variety of devices associated with IP, such as STBs, PCs, tablets, phones, etc., as well as for the different quality levels the transport network might present when connected to the end devices.
- Packaging: The packager manages the process of converting a customer request for content into a useable content stream for transport to the customer. Upon customer request, it will "package" the transcoded content data to ensure connection to the requested content is made, and that the data format is consistent with the device and associated network quality requirement.
- Edge Functions: Service providers typically use storage and/or caching strategies to buffer and manage content. These edge functions are used to ensure quality delivery of streaming video content to end-users.
- Process Management: Core functions have software applications that manage the service elements, as well as the key processes associated with delivery of video services to customers. This includes connecting the customer to the selected program, program guides, ad-insertion, etc. The intent is to ensure that key functions performed in support of delivering the services to customers via a combination of core and STB elements in legacy platforms today are mimicked in streaming services that do not require a STB.

Transcoding, packaging, edge functions as well as process management are all functions located in the data center component of the platform. The functions are implemented via a variety of high-end performance server and storage elements deployed in these facilities.

- Transport Network: Cable and telco service providers delivering linear broadcast TV services via STB-based platforms typically deliver internet services to the same customers across the same and/or similar transport network infrastructure. We note that satellite pay-TV service providers transport streaming services via internet service provider delivery platforms similar to cable and telco providers, as opposed to using the legacy satellite delivery platform.
- Customer Premises Equipment: At the customer premises, service provider internet services typically require a modem interface device. This device converts signals from the transport network into standards based ethernet signals for delivery to the devices and/or TV apps in the home used to play video. The smart TV in the home provides the capability for the customer to interact and manage channel selection and viewing via its app.

Figure 9 shows the overlay of the video streaming process flow on top of the platform.

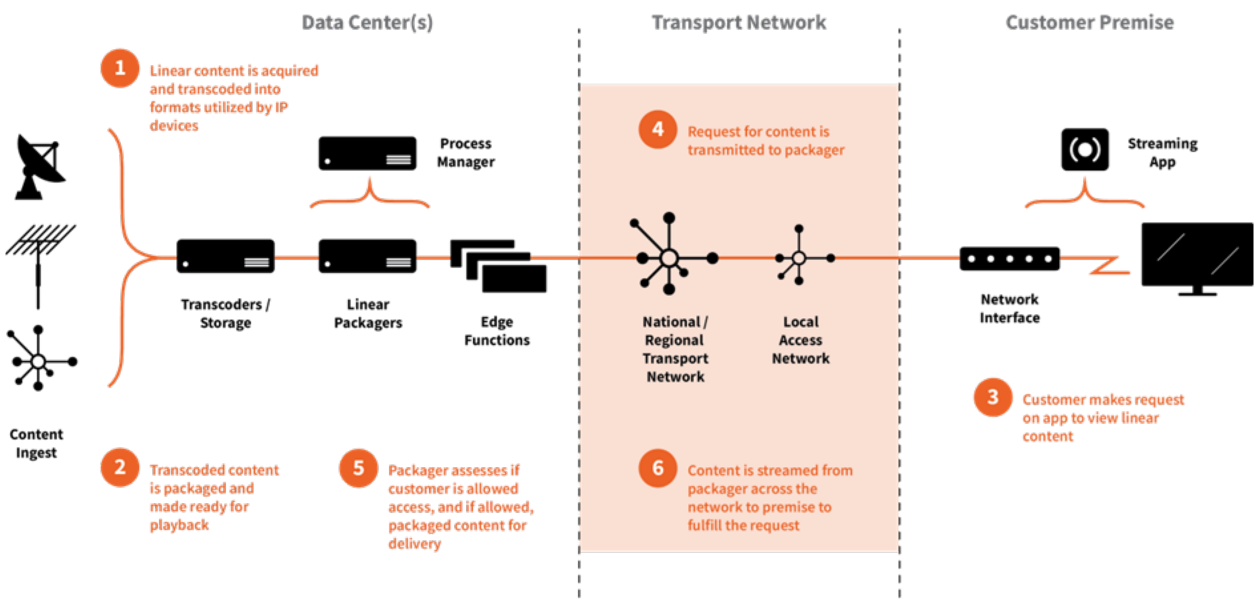


Figure 9 - Video Streaming Linear TV Process Flow

As with legacy linear TV viewing, content is ingested and processed at data centers. The process then deviates from legacy, with the content transcoded into the variety of formats IP devices use to consume video content. The content is then packaged and made ready for consumption. At the premises, a customer makes a request to view content through the streaming app deployed in the home. The request is transmitted via the network to the packager. The packager connects to the content and, assuming the customer is allowed to view the content, delivers it to the premises for viewing on the television set in the home.

3.1.4. Linear Content Record via Cloud DVR

Cloud-based DVR (cDVR) services move the linear content record and playback function of the DVR in the premises to the cloud. In cDVR, all functions associated with recording, storing, and playing back linear content for subscribers are performed centrally in elements located in a service provider's data center, hub, and/or head-end facilities.

Figure 10 shows a generic platform for service provider delivered cDVR as used in this study.

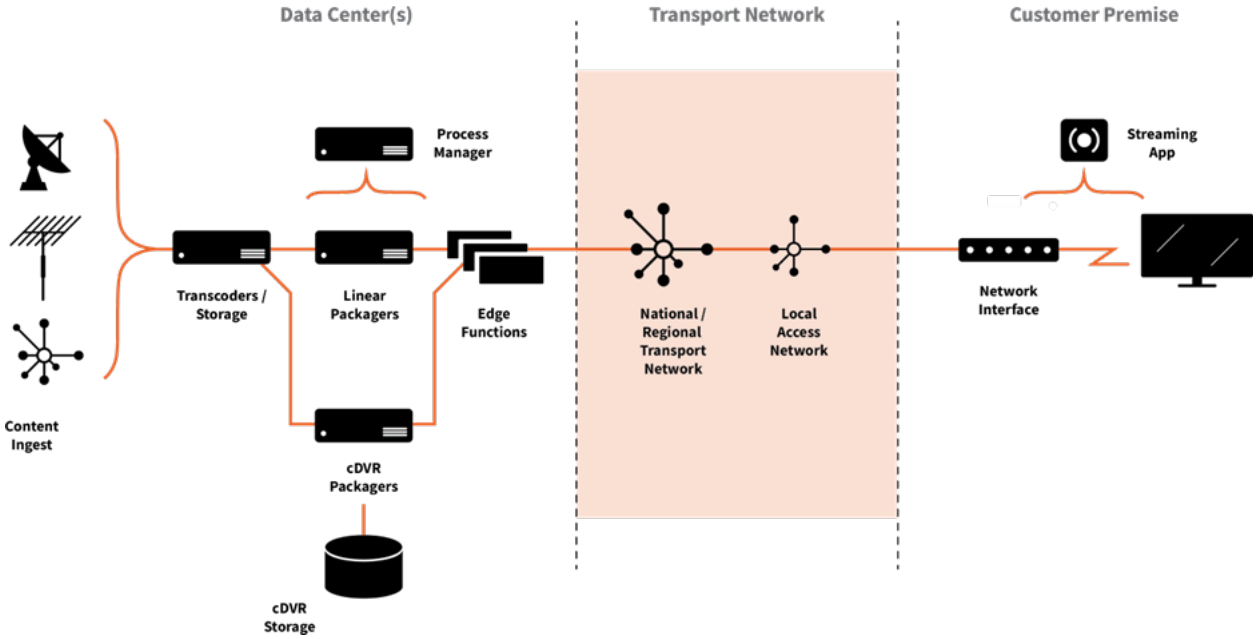


Figure 10 - cDVR Platform

As with its legacy equivalent, cDVR is overlayed on the platform utilized for video streaming delivery of linear TV content to the premises. cDVR utilizes the same content ingest and transcoded content developed for video streaming. Incremental functions added to support cDVR are related to the packaging and storage elements. For the cDVR delivery method, the packaging process is managed to ensure the record request is properly packaged and stored initially. This includes storage of content in compliance with the service provider's storage agreement with the content copyright holder. This would be either as a stored individual copy for the subscriber (Private Model), or as a request to access a stored copy of the content that is shared amongst all who request access (Shared Model). When playback of the content is requested, the cDVR packager ensures the content is transported in a format consistent with device and available network quality, like linear TV packagers. Use of the network to transport content, as well as the move of data to/from the streaming app to manage the process mirrors the streaming video case. The process is managed via an app on the TV in the home - as its name suggests, cDVR does not require storage in the home.

Figure 11 shows the overlay of the cDVR record, store, and playback functions on the platform.

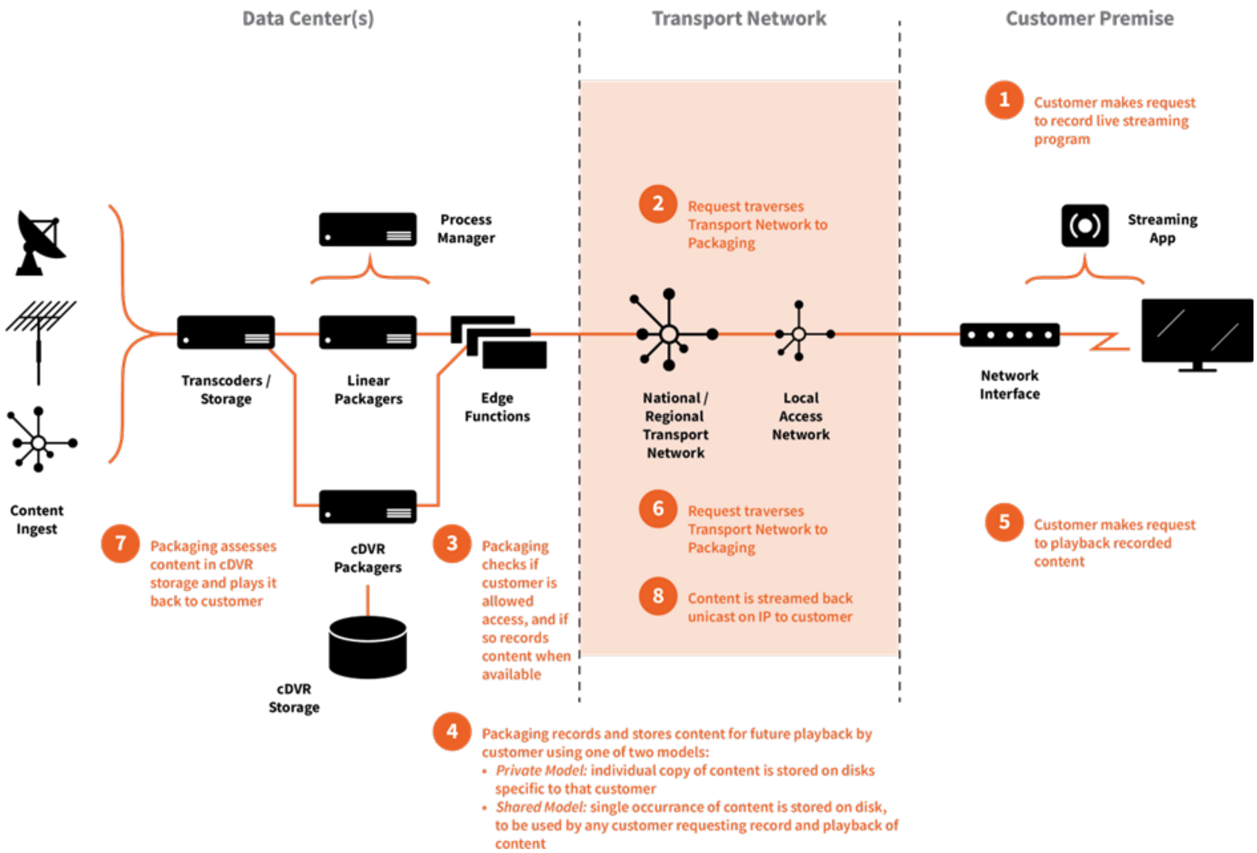


Figure 11 - cDVR Record, Store, and Playback Process Flow

From a process perspective, customers make requests to record a program in their streaming app. When they do that, the request traverses the transport network to the packager. The packager logs the request, checks permissions, etc. on the content requested, and if the customer is allowed manages the process of recording the content, ultimately storing the content consistent with the appropriate copyright agreements. As with the cDVR record function, cDVR playback is initiated by the customer in the app. When the customer requests playback, the request is transmitted across the transport network to the packager. The packager finds the content and packages it appropriately for playback on the requesting device. It then streams the content across the transport network to the customer.

4. Platform Comparison for Data Analysis

Developing the energy comparison between the two alternatives for each of the services is done through comparison of the energy usage for the component elements of each of the platforms. The focus is specifically on the component parts of the platforms where differences exist in energy usage. The sum of these differences in energy usage between the two alternatives constitutes the net energy difference between the legacy delivery method and its streaming/cloud equivalent. In each case, the legacy delivery method is viewed as the base case and/or "as is" approach. The analysis focuses on how the streaming/cloud alternative delivery method compares in energy usage to the legacy base case.

The platform comparison structures for each of the two services being compared are discussed in the following section.

4.1. Legacy vs. Streaming Service Delivery Comparison Structure

Figure 12 shows the network models of the legacy TV service delivery by STB or DVR (top) versus delivery of TV services via video streaming/cloud DVR (bottom).

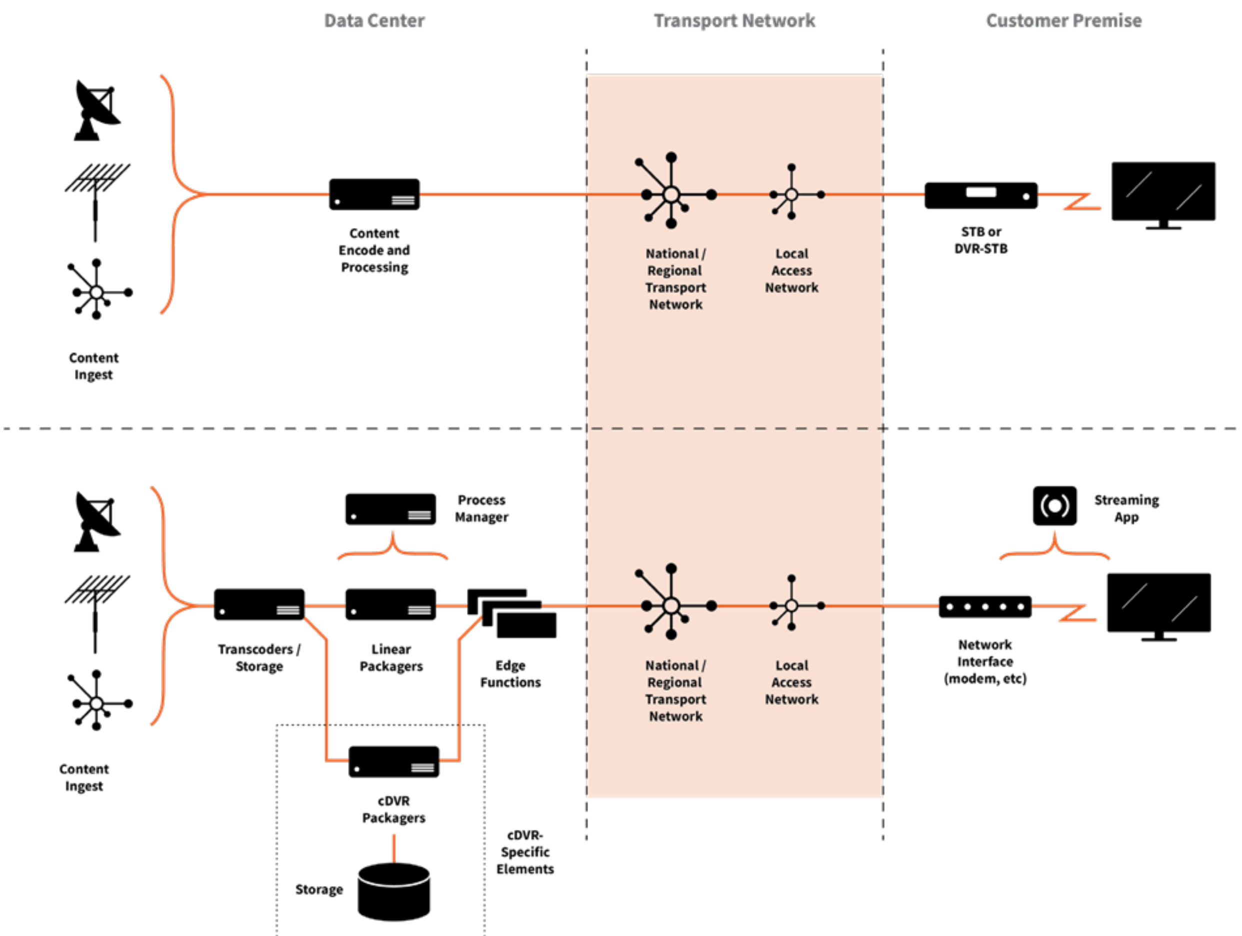


Figure 12 - Linear TV Service Delivery Platform Comparison

We analyze the component areas with respect to differences in energy usage between the alternatives. Any observed energy difference is then included in comparative calculations. If there is no or only a minimal difference in the components between the alternatives, the energy difference is assumed to be negligible. A comparative analysis of the component areas of legacy linear TV delivery method and the video streaming/cloud alternative is discussed below.

Content Ingest: The content universe ingested by a service provider for legacy TV delivery method and streaming TV delivery method is largely the same. As such, content ingest is neutral in relation to the energy comparisons being made between legacy and streaming options, as there is little or no difference in energy between the two. The energy difference is assumed negligible for this component.

Data Center: Legacy TV content processing centers primarily on processing content for efficient transport of broadcast video in the network. The video streaming delivery method requires transcoding, packaging, edge functions, and process management specific to IP based video service delivery. As these functions and the elements associated with them are additions specific to the streaming video service, energy usage for the elements is included in the analysis.

Like streaming, cloud DVR capabilities add specific functions and the associated elements required to support the service in the service provider data centers. As with their streaming counterpart, these cDVR specific elements are not a part of the legacy processing function, and as such, the energy difference for the functions are included in the analysis.

Transport Network: For legacy TV, video content traverses the transport network as a bitstream included in the linear video delivery, connecting to the STB at the customer premises. When the customer selects a channel for viewing, the STB transitions the bitstream of the selected channel, connecting it to the television for viewing. Although in streaming there are two-way interactions between the streaming app and the data center platform related to control and management of the streaming sessions that are managed locally by the STB in the legacy use case, these transmissions are "negligible" according to [10]. The bulk of the transmission is the video payload of the content being viewed. No matter if it is in the legacy network or is being streamed in IP, the content is traversing the same IP core and IP/QAM based access network, and the energy needed to transmit the content across the network in either case is similar. Thus, the energy difference for transport network is considered negligible in the analysis.

Similarly, for the TV record, store, and playback use case, the bulk of the transmission package is the video content. In the legacy DVR case, it traverses the transport network as a bitstream included in the broadcast linear video delivery and is recorded locally by the DVR at the customer request for playback at a later time. In the cDVR use case, the content is recorded by the data center component. When a customer requests playback, the content is transported at the time of the playback request for viewing. As with the streaming use case examples, it is a similar video content bitstream package transmitted across the network in both cases, just at different times. As it is being transmitted in both cases over similar physical infrastructure, it is assumed that the energy required to transport the video content package is similar in both cases. As such, the energy difference for the transport network is considered negligible in the analysis.

As noted earlier, where satellite is used to deliver content for legacy service delivery methods, the streaming delivery method is via non-satellite transport platforms like those used by internet service providers. Although energy usage for satellite uplink/downlink facilities to transport legacy delivery method video services can be quite low, less than 1% of total energy as noted in [1], the analysis in [1] also indicates that total energy consumption per hour between satellite and cable legacy delivery methods is approximately the same. Given that VA data indicates similarity between STB/DVR devices for cable and satellite delivery platforms, this implies other transport network element energy usage balances out between fixed and satellite network platforms, supporting the view that the energy difference for the transport network can be considered negligible.

Network Interface: For service providers with existing legacy STB based video delivery services, data in [3,9] indicates that broadband internet is already in place for 93%+ of the company customer relationships. Other providers require customers to have their own internet on which the streaming service is placed "over-the-top," enabling streaming service to be delivered on top of existing internet connectivity. Additionally, as noted earlier, [1] states that domestic modems and Wi-Fi routers "operate at relatively constant power consumption that vary little with respect to workload," based on typical video streaming traffic. As such, given the assumption that streaming overlays existing internet service in the home and the fact that streaming's inclusion in the throughput does not change energy usage, network interface devices in the home are considered out of scope for this analysis.

Customer Premises Equipment (CPE): The STB and DVR devices in the home are specific to legacy STB and DVR delivery method – they are not a part of the video streaming or cDVR delivery methods. As they do represent a difference in energy between the delivery methods, energy usage for STBs is included in the analysis.

End Devices: As the comparison is specific to viewing of the video content on like-for-like devices in the home such as TV sets, it is assumed that the devices used are not different enough to be relevant to the comparison. As such, end use device energy is considered out of scope for this analysis and not included. As noted in the section 2, [2] shows that some of the devices used for viewing app-delivered content such as phones and tablets generally use less energy than television sets. As such, excluding them does not create unfair bias, but instead creates potential for improvement as they are incorporated in any future analysis. There may be some energy variation between smart TVs that have direct access to apps and TVs without that capability, but for reasons independent of pay-TV apps, smart TVs are becoming the market leader in new TV sales and now represent a majority of all TVs in use in the United States.

Table 1 summarizes the component difference assumptions as used in the comparative analysis.

Table 1 - Component Comparison Summary

Component	Comparison	Comment
Content Ingest	Energy difference is assumed negligible for this component	Same content in both alternatives
Core Data Processing	Energy difference for the server and storage elements is included in the analysis	Transcode, Packaging, Edge Functions all incremental functions for streaming/cloud
Transport Network	Energy difference is assumed negligible for this component.	Same content transmitted across same/similar cable/telco network or satellite network with same/similar energy profile
STBs	Energy difference for the components is included in the analysis	Specific to legacy case, but not in streaming/cloud
Network Interface	Out-of-scope	Streaming assumed overlayed on existing internet network
Network Interface and Customer Devices	Considered "out of scope" as comparison assumes same TV set used in legacy and streaming/cloud cases, so difference will be zero regardless	Comparison assumes delivery to same device (TV set) in each implementation

5. Data Used in Study

For the study, the four largest video service providers agreed to confidentially provide energy data for streaming and cDVR usage on their respective platforms. To maintain confidentiality, this paper only reports summary data. In some cases, the company-provided data lacked certain elements required to be comparable to the other submissions. In these cases, we used publicly available data to normalize the submittals.

5.1. Service Provider Data

We requested data for calendar year 2020 to cover the incremental elements in the data center functions for streaming video and cDVR: transcoding, packaging, cDVR specific functions as well as edge functions (see Figure 12 and Table 1). Our request covered overall energy requirements as well as details on the number of customers served.

The responses we received took on various forms. For streaming, we received data from three of the four participating service providers. One company provided annual equipment level energy details, including the appropriate associated facility power usage effectiveness (PUE). The other two providers supplied annual facility level energy data in a combination of kWh for owned facilities and vCPU numbers for outsourced services. The kWh reported are at the facility level and inherently include the facility PUE. All three companies supplied the number of customer devices associated with the streaming service. For the cDVR case, we received data from all four participating companies. Three companies reported in the same format as discussed above for streaming. The fourth company provided us only with the energy requirement for one hour of cDVR usage.

We analyzed the data received and transformed the various submittals into a common format, representing the annual incremental energy per customer device for both streaming and cDVR use cases. This simply represents total annual kWh for the incremental elements divided by the number of customer devices associated with the services. To achieve this homogenization, in some cases we had to augment the reported data with data from the public domain as discussed below.

5.2. Publicly Sourced Data

As some companies reported energy as data center vCPUs (residing within Amazon Web Services), we used the reported Elastic Compute Cloud instances (Amazon EC2) and associated vCPU conversion factors and mapped these into power values using the data reported in [5]. Coupled with the reported vCPU hours, this allows us to calculate annual kWh values from the data received.

For the one cDVR data point where only data for a single hour of usage was reported, we converted this to an annual value representing a typical customer. Nielsen [6] reports viewing hours per day and per adult in the US for both linear and time-shifted (recorded) TV. We extrapolated these numbers to household values using US census data [7]. This results in annual viewing hours per household of 411 hours for time shifted TV. We were unable to differentiate the time-shifted viewing further into e.g. individual recording and on-demand viewing and are allocating the value completely to cDVR usage. This approach will introduce some error as the actual cDVR usage for this service provider may differ from the Nielsen average. We note that the calculated annual energy requirement per customer device is on the high side when compared to the other three providers.

5.3. Customer Premises Equipment for Legacy Use Cases

To compare the relative efficiency of streaming and cloud-based recording, we need to establish the baseline for legacy viewing and recording of digital video. The energy requirements of fielded set-top boxes and customer premises DVR devices are published annually under the “Voluntary Agreement for

Ongoing Improvement to the Energy Efficiency of Set-Top Boxes". The latest version [4] includes equipment fielded through 2020 and estimates purchases remaining in the field going back to 2014 or 2015.

For an overall assessment of the effectiveness of cloud-based solutions, we use the weighted average of fielded legacy equipment for comparison. This is illustrated in Table 2 and Table 3 where TEC indicates Typical Energy Consumption.

Table 2 - Fielded Non-DVR Set-Top Boxes, Typical Energy Consumption and Weighted Average [4]

	2014	2015	2016	2017	2018	2019	2020
Non-DVR Purchases from Each Year Remaining in Field	10,612,281	10,977,499	11,535,694	15,390,556	10,066,928	8,319,044	10,537,923
Non-DVR TEC Average (kWh/yr)	103.3	92.6	85.6	90.8	91.8	74.1	49.0
Weighted Average 2014-2020 (kWh/yr)	84.6						

Table 3 - Fielded DVR, Typical Energy Consumption and Weighted Average [4]

	2015	2016	2017	2018	2019	2020
DVR Purchases from Each Year Remaining in Field	7,540,600	11,219,933	8,268,205	6,304,346	5,848,219	1,719,840
DVR TEC Average (kWh/yr)	170.6	161.3	142.9	138.7	134.4	144.8
Weighted Average 2015-2020 (kWh/yr)	151.3					

5.4. Data Compilation

The results for streaming energy usage per year are shown in Table 4 on a per customer (device, e.g., TV) basis.

Table 4 - Incremental Streaming and cDVR Energy Requirements (kWh/year) per customer

	Average	Standard Deviation
Streaming	3.6	3.1
cDVR	14.4	4.7

To preserve the confidentiality of the individual contributors, we are reporting the variability in terms of standard deviation rather than minimum and maximum values. We note that there are no significant outliers outside the reported range.

The data reported represents a snapshot in time for the facility infrastructure and number of customers served reported. As customer numbers change and/or infrastructure is built out to accommodate growth, the results may change based on the balance of powered infrastructure to participating customers. This balance of infrastructure capacity vs associated customer base in this snapshot will vary between participating companies and contribute to the variability shown.

The variability between respondents is further influenced by a variety of factors not part of the responses. These include, for example, uni- vs multicast streaming percentages and storage methods (private vs shared) for the cDVR use case.

6. Summary of Results

Based on the preceding discussion of service provider data and legacy customer premises equipment, Table 5 compares the ranges of streaming/cDVR incremental energy with the legacy viewing and recording through a single set-top box or DVR.

Table 5 - Incremental Streaming and cDVR Energy Ranges vs. Legacy STB Viewing (kWh/year)

	Average minus one standard deviation	Average plus one standard deviation	Legacy Equipment (STB or DVR)
Streaming	0.5	6.7	84.6
cDVR	9.8	19.1	151.3

This allows us to establish ranges of potential annual energy savings when comparing the legacy delivery method utilizing customer premises-based STB and DVR devices to its streaming and cloud-based alternatives. Table 6 shows potential annual energy savings ranges within plus and minus one standard deviation from the average incremental energy as reported by the service providers.

Table 6 - Energy Savings — Streaming and cDVR vs. Replacing Legacy STB (kWh/year)

	Average minus one standard deviation	Average	Average plus one standard deviation
Streaming	78.0	81.1	84.1
cDVR	132.2	136.9	141.5

As illustrated, the energy savings for cloud-based services over premises-based solutions can be significant. When compared to the traditional use of a non-DVR set-top box, streaming saves approximately 81 kWh per device per year. Delivering digital video recording through the cloud saves almost 137 kWh per device per year on average.

As the equipment energy requirements above are based on a weighted average of fielded equipment, the actual difference would depend on the vintage of equipment being displaced and/or migrated. For example, as illustrated in Table 2, replacing an older STB increases the energy savings above the value listed in Table 6, while replacing a newer STB will lead to lower energy savings than listed. No matter which vintage of end use device is being replaced, the end result of replacement would always be net energy savings.

It must be noted that the calculations shown in Table 6 are based on only one of many potential scenarios: the elimination of a single STB or DVR, respectively. In reality there are many potential variations. A household might choose to replace multiple devices with streaming, resulting in greater savings. If a consumer uses streaming (such as to watch video on a tablet) but still chooses to retain their STB/DVR, the energy reduction shown in Table 6 is not achieved, but the energy usage of the end use device will be reduced as it will more often be in a reduced-power mode. In other cases, customers may retain non-DVR set-top boxes for linear TV but utilize recording functionality via cloud DVR, in which case the energy savings will also be reduced from the complete replacement assumption underlying the results in Table 6.

In addition, the savings shown assume that the required streaming apps are provisioned in the viewing device (e.g., tablet or smart TV). If a separate device is required to access streaming functions, the associated energy requirement will increase streaming energy usage, and lower the savings shown. Although the focus of our work was to quantify potential savings offered by streaming/cloud approaches in a STB/DVR complete replacement scenario, which we show as significant, understanding actual savings for the different migration strategies would require further analysis.

Continuing the movement of linear and recorded television content viewing from the legacy STB/DVR based delivery method to the streaming and cloud-based alternatives will lower energy usage in the home, as well as the related carbon emissions. From a sustainability perspective, it makes sense to continue this evolution in the future.

7. Observations and Future Improvement

The calculated energy differential per subscriber in Table 6 represents energy usage at a snapshot in time. Evolution of the streaming/cloud-based delivery method platforms as subscriber usage increases and as platform components, functions, elements, and devices become more efficient, has the potential to impact the comparison in the future. The following sections detail potential opportunities for efficiency improvement and discuss the impact of platform efficiency evolution.

7.1. Facility Efficiency

We noted that the power usage effectiveness (PUE) values for the facilities reported by the service providers vary significantly. The average PUE observed is approximately 1.5. All the incremental energy associated with streaming and cloud-based services is facility based. Thus, the impact of improved PUE values flows linearly to energy savings. An improvement of PUE by 10% would result in 10% less energy required and a commensurate improvement in energy savings for the streaming/cloud-based platforms, making the comparisons more favorable towards those alternatives.

7.2. cDVR Implementation

For cDVR services, the nature of the implementation has a large impact on the energy requirement. There are two fundamental ways of establishing cloud-based DVR systems. In the "private content storage model" a unique copy of content is stored every time a user records the content. In the "shared copy storage model" a single copy of a program is made and shared by all subscribers who have recorded the program. The service providers participating in this study did not share the exact details of their respective cDVR implementations.

Implementation of these models depends on the individual agreements with the respective content and copyright owners. As a result of interpretations of copyright law and contracts, cDVR services in the United States have typically employed the "private model" for at least some content. This method is very storage intensive, and the requirements scale linearly with the number of customers. It has been reported [8] that the cost of ownership associated with a full private model is a factor of 10 above a shared copy model (or a private copy with de-duplication).

The status quo today is likely a mix between private and shared models, as some content owners may have agreed to the flexible right, but others have not.

It is understood that moving towards a full implementation of a shared model is subject to legal considerations and negotiations with copyright owners. From an energy savings perspective, however, the anticipated benefits would be significant. We believe that energy usage improvements of 50% or more beyond the cDVR range reported in Table 4 are well within reach. These energy savings would further improve cDVR performance when compared to the legacy approach.

7.3. Energy Impact of Data Center Component Evolution

Energy usage is driven by the equipment service providers deploy to produce the platform. Data center components consist primarily of server and storage equipment, either owned or housed in third-party data centers. The applications performing the functions are run on the servers, utilizing storage as and when needed.

Initially, equipment for the content ingest and transcode functions is sized based on the channel capacity to be delivered. Although channel capacity differs by service provider, for streaming service providers it is typically in the hundreds to low thousands of the channels streamed. Adding equipment in these areas is driven by the need of the service provider to add channels.

Equipment elements for streaming, transcoding, and packaging functions are built initially based on subscriber numbers and usage assumptions service providers make. Growing the platform is typically via tranches of these equipment elements added in step functions. The element steps typically add capacity to cover subscriber and usage growth for a reasonable period (e.g., 1-3 years), while not placing so much as to have site capacity unused for a long time.

Similarly, cDVR packaging and storage is initially sized based on subscriber and usage assumptions. This includes assumptions around storage requirements based on the cDVR model used as discussed in 7.2. Usage growth drives equipment and storage element additions to the platform, similarly incremented in capacity tranches balancing the need to stay ahead of customer requirements versus operational and capital efficiency.

As noted earlier, the calculated energy per subscriber for the data center component represents a snapshot in time. As service providers scale the platform, energy will fluctuate as tranches of equipment elements are placed for additional channels and/or subscriber usage. Although service provider data provided did not give a quantitative view, it is expected that, as the platform grows, energy per subscriber will fluctuate somewhat around the snapshot numbers shown. These fluctuations should have minimal/no impact on any future view of this analysis in the event the energy usage snapshot is taken at a different time — we expect that any snapshot will have an output in a comparable range for energy per subscriber as presented here. Scaling up the platform should not impact the analysis in a meaningful way.

7.4. Energy Impact of STB Evolution

As shown in Table 1 and Table 2, STB and DVR energy is calculated as a weighted average based on STB/DVR implementation over time. As the tables show, outside of the 2020 DVR result, the average energy for the devices has been reducing meaningfully over the last few years. Based on a combination of continued year-on-year reduction in STB and DVR energy and the replacement of older devices with newer ones, we expect a continued decrease of the weighted average energy over time.

As shown in Table 3, the magnitude of the savings is significant. Although the continued reduction of STB and DVR energy will continue to narrow the savings, reduction in device energy usage will not change the outcome – energy usage of in-home STB and DVRs would be expected to remain higher than energy usage of streaming and cDVR for video content delivery.

8. Summary and Conclusions

The analysis presented is based on data provided by four signatory companies to the “Voluntary Agreement for the Ongoing Improvement to the Energy Efficiency of Set-Top Boxes”. We parsed the data to identify power components related to streaming and cloud-based services that are additive when compared to the legacy approach. The results are then compared to the powering characteristics of customer premises-based STBs required for legacy viewing and video recording.

We find that video streaming and cloud-based recording can reduce the energy requirement significantly, by an average of 81 kWh per year and 137 kWh per year, respectively. We do note that the listed savings may be impacted by other factors, such as for example the age of the legacy equipment, or the migration strategy associated with moving from legacy STB/DVR to streaming/cloud. Not only do the cloud-based approaches require less energy, but they also enable customers to use existing equipment (e.g., phones or tablets) to stream or record/play video not only in the premises but wherever a connection is available. Although out of scope of this study, it is worth noting that viewing on devices such as phones and tablets generally uses less energy than viewing on a television set.

We believe that the energy required for cloud-based solutions can be reduced further. The equipment elements required for implementation are generally housed in facilities and data centers. Improving the efficiency of these facilities translates directly to reduced energy requirements for streaming and cDVR operation. Cloud-based video recording in the United States is still very reliant on a private copy model, where an individual copy is made every time a customer records content. Sharing a single copy of a program between users is significantly more energy efficient by reducing the processing and storage hardware requirements and associated costs. To date, copyright considerations have constrained a more widespread implementation of this sharing approach. From an energy efficiency perspective, it is hoped that legal and contractual solutions can be found in the future, since the energy savings benefits are significant.

Based on the efforts of the parties to the voluntary agreement, STB device energy is likely to decrease in future years. This may narrow the energy gap between legacy STB based viewing and streaming or cloud-based approaches. However, we do not believe that the legacy approach will ever reach parity with or be more efficient than the cloud-based implementations. This is especially true in light of the streaming improvements discussed in Section 7. The migration of television viewing on home television sets from legacy STB delivery to its streaming and cloud-based alternatives will create benefits by lowering energy usage and reducing the associated carbon emissions. From a sustainability perspective, we support a continued and speedy implementation.

9. Abbreviations

BBC	British Broadcasting Corporation
cDVR	cloud-based DVR
CPE	customer premises equipment
DVR	digital video recorder
FTTH	fiber-to-the-home
HD	high definition
HFC	hybrid-fiber coax
IP	internet protocol
ONTs	optical network terminations
PUE	power usage effectiveness
QAM	quadrature amplitude modulation
QPSK	quadrature phase shift keying
SD	standard definition
STB	set top box
vCPU	virtual central processing unit

10. References

1. Schien, Daniel et al, “Using Behavioural Data to Assess the Environmental Impact of Electricity Consumption of Alternate Television Service Distribution Platforms”, BBC Research & Development White Paper 372, September 2020. (link)
2. Carbon Trust, “Carbon Impact of Video Streaming”, June 2021. (link)
3. Comcast Corporation, Form 10-K, 2021 Annual Report. (link)
4. D+R International, 2020 Annual Report, “Voluntary Agreement for Ongoing Improvement to the Energy Efficiency of Set-Top Boxes”, published August 23, 2021. (link)

5. Benjamin DAVY, “Building an AWS EC2 Carbon Emissions Dataset”, Teads Engineering, published September 21, 2021. (link)
6. The Nielsen Company, “The Nielsen Total Audience Report”, published March 2021. (link)
7. U.S. Census Bureau, Current Population Survey, Annual Social and Economic Supplements, 1940 and 1947 to 2021. (link)
8. Pieter Liefooghe , Broadpeak, “Blog Series: Navigating Cloud DVR’s Murky Legal Waters – Part 2”, June 10, 2021. (link)
9. Charter Communications, Form 10-K, 2021 Annual Report. (link)
10. CableLabs (2020, July 30). Upstream: How Much Speed Do You Need? InformED Blog (link) Note: Scroll to date on the blog link to find reference entry

Aluminum Cable Applications With Energy Storage In Cable Head Ends And Data Centers

A Technical Paper prepared for SCTE by

Carly Waschka, Product Manager, Priority Wire & Cable, SCTE Member
1800 E. Roosevelt Road
Little Rock, AR
cwaschka@prioritywire.com
346-444-9231

Larry Hamilton, Advisor, Priority Wire & Cable
1800 E. Roosevelt Road
Little Rock, AR
larryh@prioritywire.com
303-638-1446

Table of Contents

Title	Page Number
Table of Contents	32
1. Introduction	33
2. Copper and Aluminum Cables in Broadband Energy Storage	33
2.1. Industry Trends Since 2000	33
2.2. Copper and Aluminum Conductivity, Weight, Cost	34
2.3. Sample Power Plant / Battery Plant Power Cable Economics	35
2.4. 8000 Series Aluminum Performance	36
2.5. Aluminum gains in environmental impact	37
3. Conclusions	38
4. Bibliography and References	39

List of Tables

Title	Page Number
Table 1 - Solar Installations with Aluminum Cabling	34
Table 2 - UPS Aluminum / Copper Cable Sizing Chart, Global Supplier	34
Table 3 - NEC AMPACITY CHART 310.21	35
Table 4 - Power Plant Backup Power Economics	36
Table 5 - AL / Copper Analytics	36
Table 6 - Creep Loss / Cold Flow Comparison	37
Table 7 - Environmental Impact – Aluminum	38

1. Introduction

Power systems in the US have undergone very little change over the past century. We still generate, distribute and store power in much the same way. However, we are moving to better models. Solar and wind are replacing fossil fuels. We are making inroads on micro grids and newer technologies like hydrogen. SCTE Energy 20/20 is taking an industry through the phases of analysis and growth.

Part of the power delivery system is the choice of conductors and connectors. The power transmission systems – from power plant to substation – switched to aluminum from copper decades ago, including DC transmission like Bonneville Power. Most power distribution infrastructure is now aluminum, as are most service drops. The solar industry – the collection systems, using 1or 2kV power cable – switched to aluminum as well. The decision was simple:

- Weight reduction for cable. Savings of over 40% on weight, even after an adjustment in size for conductivity.
- Cost. The cost of aluminum/pound is less than 40% of the cost of copper/pound.
- With 8000 series alloys, the traditional concerns of cold flow are now minimized. Cold flow refers to aluminum going out of compression termination, which then causes a loose connection that overheats.

The use of copper for battery and power in the cable broadband provider market is still prevalent. Significant savings in freight, CAPEX, and even carbon credits are in play with a switch to aluminum. Today, the broadband supply chain stocks AL THHN; XHHW; RHH/RHW in multiple sizes and colors. The current practice – and a potential change - is worthy of the SCTE Energy 20/20 team's consideration and review.

2. Copper and Aluminum Cables in Broadband Energy Storage

2.1. Industry Trends Since 2000

The power backup systems of 2000 and 2022 are not significantly different. While the industry has made spot usage of solar and wind for backup power and can buy both from most power utilities as sourcing mechanisms to support green primary power, the typical power backup system continues to be an internal combustion engine-based generator with battery plants – and a significant number of broadband utilities use redundant power and battery plants for resiliency. The power generation industry has adopted large solar implementations with a major shift to aluminum from copper on collector system voltages. Table 1 shows a national footprint of solar farms that have used 8000 series aluminum – total installed value is well over $1B of aluminum cabling.

Table 1 - Solar Installations with Aluminum Cabling

In addition, major manufacturers have started to add aluminum sizing options to their usage guides. See Table 2:

Table 2 - UPS Aluminum / Copper Cable Sizing Chart, Global Supplier

UPS with 1000 kW I/O Cabinet 480 V — Specifications

Recommended Upstream Protection and Cable Sizes for 1000 kW UPS

	Maximum OCPD (A)	Conductors per Phase Copper/ Aluminium (kcmil)	Equipment Grounding Conductor Copper/Aluminium[10]
Input	1600[11]	4x600 / –	4/0 AWG / –
Bypass	1600[12]	4x400 / –	4/0 AWG / –
Output	1600[12]	4x400 / –	4/0 AWG / –
Battery	3000[11]	8x500 / –	400 kcmil / –

2.2. Copper and Aluminum Conductivity, Weight, Cost

The core criteria of the cable materials decision follow performance and economics – and legacy practices. Later information will detail performance; the basics of economics are driven by copper and aluminum metals pricing and the weight per pound per foot of each cable used in an identical design model. The NEC Ampacity Table for Aluminum and Copper (See Table 2) defines the model. Aluminum conductivity is less than copper; as a result, depending on size and NEC chart definitions, aluminum design must increase 1 or 2 sizes to match copper ampacity. As a result, in conduit environments, % fill must be managed against code requirements. Given than most head end and data center power infrastructure deployments are in tray, this restriction is not considered a limitation.

Table 3 - NEC AMPACITY CHART 310.21

ARTICLE 310 — CONDUCTORS FOR GENERAL WIRING 310.21

Table 310.16 Ampacities of Insulated Conductors with Not More Than Three Current-Carrying Conductors in Raceway, Cable, or Earth (Directly Buried)

Size AWG or kcmil	Temperature Rating of Conductor [See Table 310.4(A)]						Size AWG or kcmil
	60°C (140°F)	75°C (167°F)	90°C (194°F)	60°C (140°F)	75°C (167°F)	90°C (194°F)	
	Types TW, UF	Types RHW, THHW, THW, THWN, XHHW, XHWN, USE, ZW	Types TBS, SA, SIS, FEP, FEPB, MI, PFA, RHH, RHW-2, THHN, THHW, THW-2, THWN-2, USE-2, XHH, XHHW, XHHW-2, XHWN, XHWN-2, XHHN, Z, ZW-2	Types TW, UF	Types RHW, THHW, THW, THWN, XHHW, XHWN, USE	Types TBS, SA, SIS, THHN, THHW, THW-2, THWN-2, RHH, RHW-2, USE-2, XHH, XHHW, XHHW-2, XHWN, XHWN-2, XHHN	
	COPPER			ALUMINUM OR COPPER-CLAD ALUMINUM			
18*	—	—	14	—	—	—	—
16*	—	—	18	—	—	—	—
14*	15	20	25	—	—	—	—
12*	20	25	30	15	20	25	12*
10*	30	35	40	25	30	35	10*
8	40	50	55	35	40	45	8
6	55	65	75	40	50	55	6
4	70	85	95	55	65	75	4
3	85	100	115	65	75	85	3
2	95	115	130	75	90	100	2
1	110	130	145	85	100	115	1
1/0	125	150	170	100	120	135	1/0
2/0	145	175	195	115	135	150	2/0
3/0	165	200	225	130	155	175	3/0
4/0	195	230	260	150	180	205	4/0
250	215	255	290	170	205	230	250
300	240	285	320	195	230	260	300
350	260	310	350	210	250	280	350
400	280	335	380	225	270	305	400
500	320	380	430	260	310	350	500
600	350	420	475	285	340	385	600
700	385	460	520	315	375	425	700
750	400	475	535	320	385	435	750
800	410	490	555	330	395	445	800
900	435	520	585	355	425	480	900
1000	455	545	615	375	445	500	1000
1250	495	590	665	405	485	545	1250
1500	525	625	705	435	520	585	1500
1750	545	650	735	455	545	615	1750
2000	555	665	750	470	560	630	2000

Notes:
1. Section 310.15(B) shall be referenced for ampacity correction factors where the ambient temperature is other than 30°C (86°F).
2. Section 310.15(C)(1) shall be referenced for more than three current-carrying conductors.

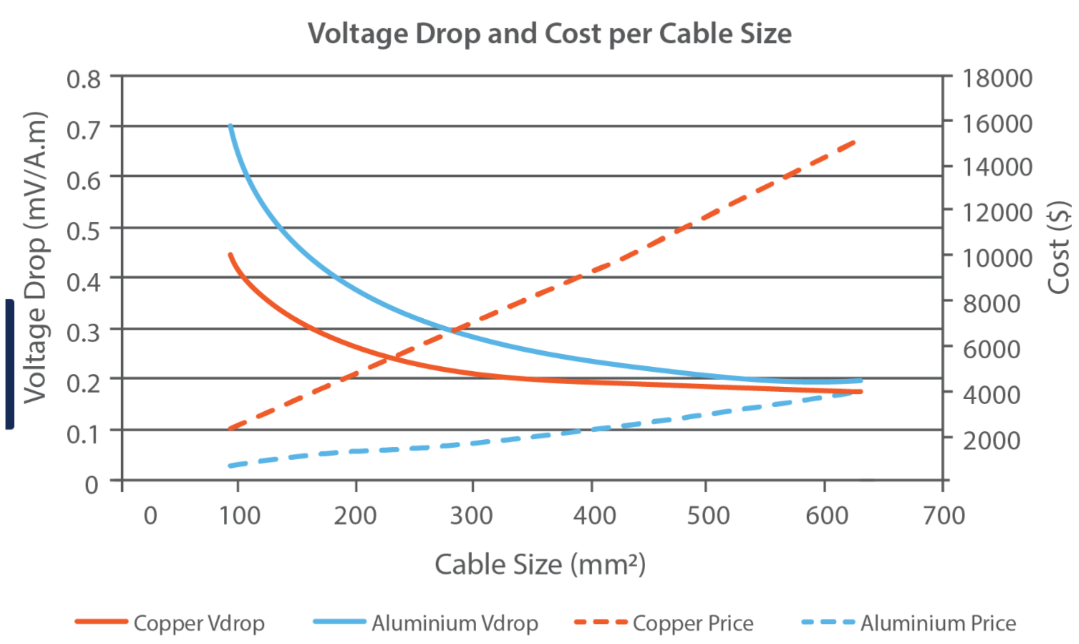
2.3. Sample Power Plant / Battery Plant Power Cable Economics

To build a sample power plant and create a real-world example, variables are minimized. A typical head end across several MSOs has some real variability; thus the intent is to focus on basic levels of distance and redundancy. See Table 4 for the results. As mentioned in the table footer, the savings associated with large reductions in weight – reducing freight and handling costs – are not defined; they are somewhat subjective and not needed for the analysis. The data is conclusive – material savings of over 80% against metals and 40-60% overall. For an analysis against both voltage drop and cost, please see Table 5.

Table 4 - Power Plant Backup Power Economics

	Copper vs Aluminum backup Power Economics					**Sep-2022**
		(Copper @ $3.56/lb; Aluminum @ $1.04/lb)				
DESIGN	A & B feeds, Generator to power plant				**DISTANCE**	100 feet from generator to power plants
	A&B feeds, power plant to battery plants					75 feet from power plant to battery plants
CONDUCTORS	3 phase 4/0 copper to power plants				300 MCM AL 8176 alloy to power plants	
	500 MCM Copper to battery plants				750 MCM AL 8176 alloy to battery plants	
COSTS	Copper	$1,395		AL	$124	
		$1,649			$253	
Totals		**$3,044**			**$377**	
	Cost savings does not include reduced freight or handling considerations					
	Savings is simply difference in metals.					

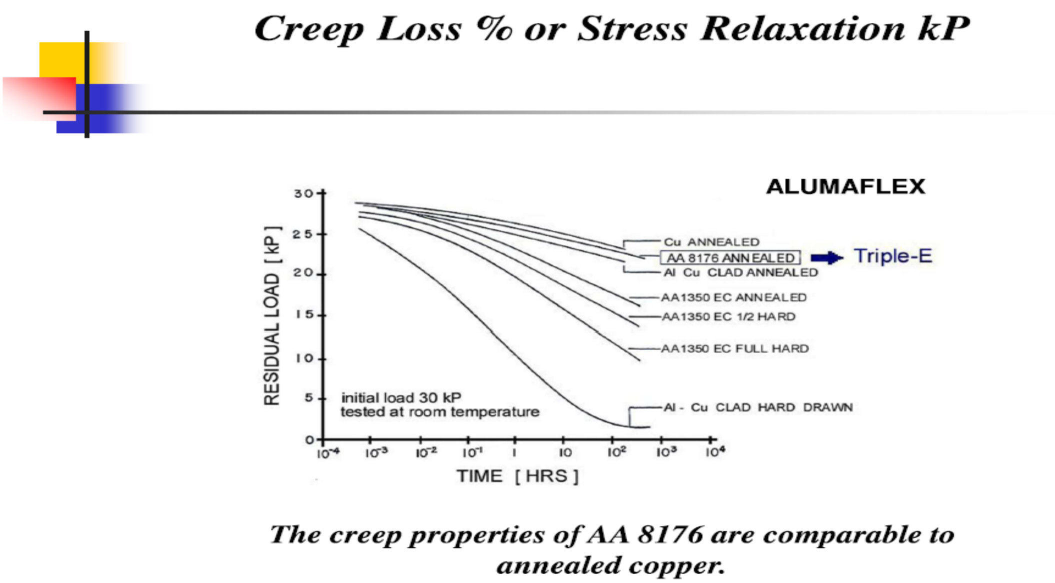
Table 5 - AL / Copper Analytics



2.4. 8000 Series Aluminum Performance

The history of aluminum building wire includes an initial deployment of aluminum series 1350 alloy for commercial and residential applications. It was a mistake – this alloy has poor cold flow characteristics in comparison to copper. For several decades, the use of 8000 series aluminum has been the standard, without any system-wide concerns. Table 6 exhibits the comparison of the most common 8000 series alloy, AA8176, with copper. The facts demonstrate that there is parity now in cold flow performance, which is why most utilities now depend on aluminum cable across their transmission, distribution, and service drop assets.

Table 6 - Creep Loss / Cold Flow Comparison



2.5. Aluminum gains in environmental impact

The aluminum industry led the way into recycling of reusable materials. Since the early 90s, the industry has continued to drive better performance (See Table 7 graphs, below). With ongoing growth of clean power, and increased recycling, this improvement is ongoing.

Table 7 - Environmental Impact – Aluminum

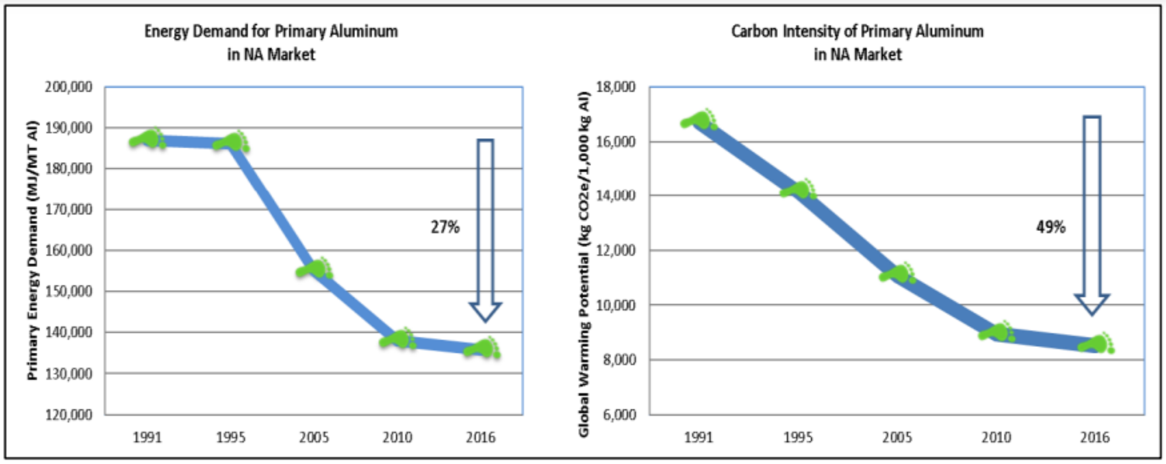
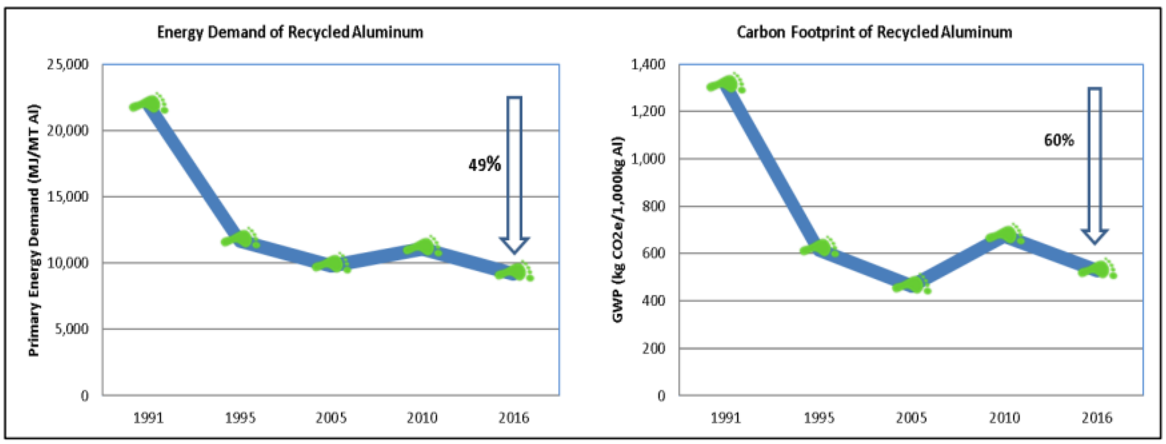


Figure 9-13: Trend of primary energy demand and carbon footprint associated with primary aluminum production.



3. Conclusions

The history of wire and cable traces back to the origins of electricity. Power / backup power systems in communications industry are now dated back to the late 1800s, when -48DC power drove copper signals to the home. Since that time, copper wire has been the mainstay of the broadband energy storage process.

However, technology and society continue to evolve and the needs have changed. SCTE created Energy 20/20 to support reduction of the industry's carbon footprint and costs and to promote use of modern methods. Just as the solar industry has converted its 600V, 1kV and 2 kV systems from copper to aluminum, the battery storage ecosystem must make similar changes to keep its standards current and cost-effective. From the analytics of performance and economics, the decision is clear – the industry gets greener and more cost effective with aluminum as the core cable for energy backup systems.

4. Bibliography and References

"Aluminum vs. Copper DC Cables; Which is Better?" Global Sustainability Energy Solutions (GSES), May 2, 2019

"The Environmental Footprint of Semi-fabricated Aluminum in North America – A Life Cycle Assessment Report", The Aluminum Association, Jinlong Wang, January 2022

"48VDC Power and the Backbone of the Telecommunications Industry", Server Technology, Annie Paquette, October 4, 2019

"UPS Sizing charts"; Schneider Electric, October 2019

Solar deployments map: Priority Wire & Cable, September 2022

"Evaluation of Aluminum Cable"; IEEE OCS, Brent Booker, Southwire, September 14, 2011

SCTE STANDARDS JOURNAL

L4S Transport Over DOCSIS

Experiments and Observations of a Low Latency Transport Protocol Over DOCSIS Networks

A Technical Paper prepared for SCTE by

Ram Ranganathan
Director, Technology & Strategy
Access Network Solutions
CommScope
Toronto, Canada
+1 905 568 7317
ram.ranganathan@commscope.com

SCTE STANDARDS JOURNAL

Table of Contents

Title	Page Number
Table of Contents	41
1. Introduction	43
2. Background on Transport Protocols	43
3. Explicit Congestion Notification and L4S	44
4. Low Latency DOCSIS and synergy with L4S	45
4.1. Classifying Low Latency Traffic	47
4.2. Queue Protection Function in LLD	47
4.3. Active Queue Management in LLD	48
4.4. LLD Weighted Scheduler	48
4.5. Proactive Grant Service	49
5. Low Latency DOCSIS Measurements	49
6. L4S on DOCSIS Testbed Setup	51
7. L4S over DOCSIS: Experiments & Observations	52
7.1. L4S baseline tests	52
7.2. AQM Coupling Behavior	55
7.3. Dual Queue Weighted Scheduler	56
7.4. Proactive Granting and L4S traffic	56
8. L4S Adoption and Industry Activity	57
9. Conclusions	57
10. Abbreviations and Definitions	58
11. Bibliography and References	59

List of Figures

Title	Page Number
Figure 1 - ECN in IPv4 header	44
Figure 2 - ECN in IPv6 header	44
Figure 3 - Typical L4S system behavior	45
Figure 4 - LLD ASF concept	46
Figure 5 - LLD Classification	47
Figure 6 - Queue Protection and Packet Sanctioning in LLD	47
Figure 7 - AQM and Coupling in Low Latency DOCSIS	48
Figure 8 - PGS granting in DOCSIS	49
Figure 9 - LLD Upstream Test Results	50
Figure 10 - QP Impact on High Bandwidth LL Stream	51
Figure 11 - L4S DOCSIS Experiment Setup	52
Figure 12 - L4S Poor throughput with 1ms Target Latency	53
Figure 13 - L4S vs Cubic Baseline Throughput with 5ms Target Latency	53
Figure 14 - L4S vs Cubic Baseline Latencies with 5ms Target Latency	54
Figure 15 - PGS vs BE Latency Comparison	56

List of Tables

Title	Page Number
Table 1 - Baseline L4S vs Cubic metrics	54
Table 2 - Impact of AQM Coupling factor	55
Table 3 - Impact of WRR Scheduling Weight	56

1. Introduction

Low loss low latency scalable (L4S) Transport is a newer internet transport technology that allows congestion feedback signals from bottleneck links such as those between a cable modem and CMTS to be signaled back to the sender, which can react to those signals rapidly and reduce queue building and buffer bloat issues. This allows protocols that typically require high bandwidth such as virtual or augmented reality applications to leverage technologies like low latency DOCSIS and avoid queue building by responding rapidly to the incoming ECN congestion signals. To effectively balance L4S traffic with classic TCP traffic, a dual-queue coupled active queue management (AQM)-based system may be used and such a dual-queue system is implemented in Low Latency DOCSIS. Our study analyzes the various parameters of L4S and how tuning those metrics impact the coexistence of L4S and classic traffic on the dual queue system. Additionally, the behavior of L4S-based traffic over a latency-optimized proactive grant service (PGS) service flow is also presented. Understanding the behavior of protocols like L4S TCP in real-world environments helps in leveraging such transport technologies for adaptive high bandwidth apps that also require low latency and jitter to respond to end-user interactivity.

2. Background on Transport Protocols

Classic TCP determines how much data it transmits over a network by utilizing a sender-side congestion window and a receiver-side advertised window. The nature of TCP is to burst to available capacity between the sender and receiver and adjust the sending window and throughput based on network behavior. The most common TCP congestion protocols in use such as Reno and Cubic use packet drops as a mechanism to determine network congestion, though other heuristics can also affect overall sending behavior. Relying on packet drops often results in a sawtooth behavior with congestion which is not ideal for low latency applications.

User Datagram Protocol (UDP) transport is typically used for send-and-forget transmissions, often associated with low-bitrate streams. Recently, UDP has been increasingly used as a base transport layer for higher-level protocols such as QUIC which provides connection management, retransmissions, and other sophisticated primitives. Video conferencing as well as streaming applications also often leverage UDP transport with their own adaptive algorithms and congestion control management at a higher layer. There have been ongoing studies on the fairness of some of these protocols when mixed with classic TCP traffic in the network.

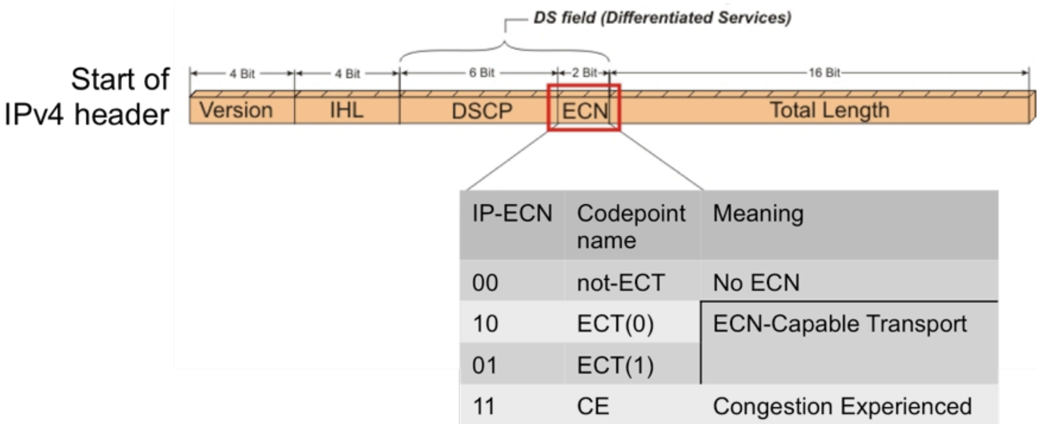
Traditionally, applications requiring high bandwidth (file transfers, video streaming) have been able to tolerate some level of latency and packet delay variation due to application-level buffering. Other latency-sensitive traffic (interactive gaming controls, sensor data, voice over IP) have been typically low bitrate and have leveraged UDP-based transport with additional protections such as lower layer QoS reservations and optimizations.

We are now seeing applications that require both high bandwidth and low latency – such as cloud gaming, interactive applications based on virtual reality and augmented reality, etc. Other applications in the same vein are likely to evolve, primarily driven by the availability of higher access network bandwidth. DOCSIS 4.0, newer PON technologies like XGS-PON and NGPON2, Wi-Fi 6 and beyond, and 5G networks all promise multi-gigabit speed to end users.

3. Explicit Congestion Notification and L4S

Explicit Congestion Notification (ECN) is a construct defined for TCP/IP that is designed to signal end-to-end congestion without having to drop packets. ECN was originally defined in [RFC 3168] and leverages the additional bits in the IP TOS header in IPv4 and TrafficClass header in IPv6 as shown in Figure 1 and Figure 2.

Marking the header to indicate the device as ECN capable transport allows middlebox systems to mark packets as congestion experienced (CE) instead of dropping them and allows endpoints to react to those signals better rather than having to rely on drops and retransmissions.



IP-ECN	Codepoint name	Meaning
00	not-ECT	No ECN
10	ECT(0)	ECN-Capable Transport
01	ECT(1)	
11	CE	Congestion Experienced

Figure 1 - ECN in IPv4 header

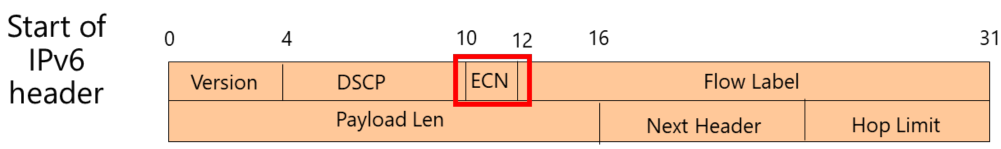


Figure 2 - ECN in IPv6 header

Traditional use of ECN based on RFC 3168 tends to use ECT (0) and studies have shown [DRAFT L4SOPS-TSVWG] that such deployments are fairly limited in the public Internet today and are often associated with FQ_CoDeL based fair queuing in middlebox systems and in some cases, single FIFO queues. In traditional ECN use, a CE marking is treated as equivalent to a packet drop for congestion control protocols.

Low latency low loss scalable throughput (L4S) transport leverages the ECN concept but allows for a more fine-grained approach to handling CE-marked traffic that is less severe than treating as packet drops. It includes three key architectural constructs as described in [DRAFT L4SARCH-TSVWG]:

1. A scalable congestion control algorithm at the transport sender that responds to ECN congestion signals. Examples included Data Center TCP or DCTCP and TCP Prague.

2. Some form of separation of L4S traffic from classic traffic in the middlebox systems. While fair queuing systems like FQ_CoDel can be used for this traffic separation, a simple dual queue system in which low latency traffic is separated from classic traffic would suffice. The classic traffic continues to rely on packet drops for congestion management
3. The protocol itself, which marks the packets with ECT bits and the middlebox calculating marking probabilities and choosing when to mark packets as congestion experienced (CE). Various AQM algorithms and improvements to AQM behavior such as "coupling" of probabilities may be employed here for ideal marking behavior.

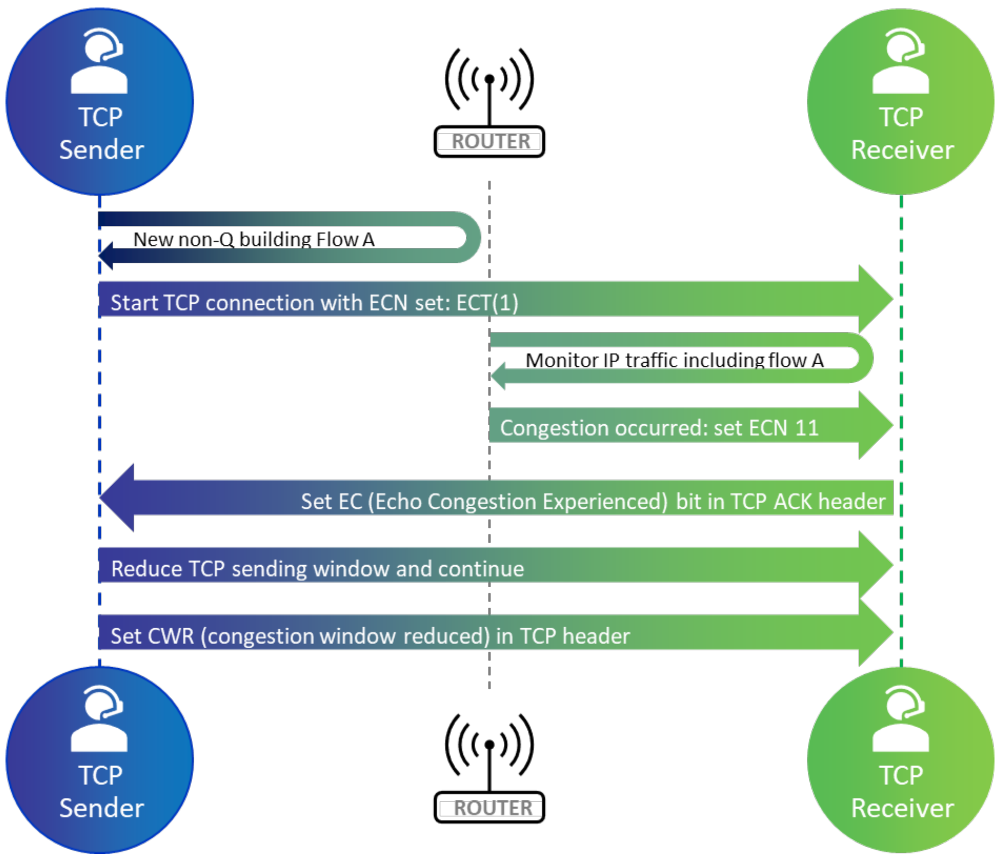


Figure 3 - Typical L4S system behavior

4. Low Latency DOCSIS and synergy with L4S

Low latency DOCSIS (LLD) is a key component of the DOCSIS specifications [DOCSIS MULPI]. Defined as part of DOCSIS 3.1 technologies and beyond, the focus is to reduce latency and packet delay variation for both upstream and downstream traffic. At the heart of LLD is the separation of application traffic between queue-building and non-queue-building (NQB) applications. NQB applications are sensitive to latency and respond poorly to queue building and buffer bloat in middlebox systems. LLD targets round-trip latencies as low as 1ms at the 99th percentile for such NQB applications.

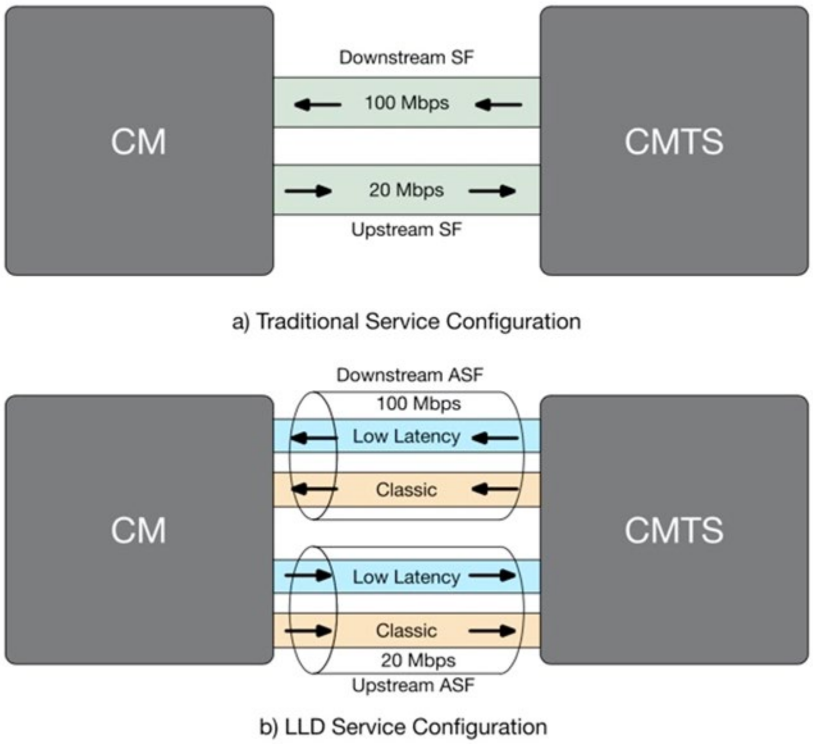


Figure 4 - LLD ASF concept

LLD achieves this traffic isolation by using the construct of an aggregate service flow (ASF) which allows low latency service flow and classic service flow to be viewed as an aggregate for DOCSIS traffic management such as rate shaping and bandwidth limit enforcement. However, separating the two service flows also allows for individual queue management and AQM algorithms for both the low latency traffic and classic traffic.

The LLD traffic separation into dual service flows is very similar to the dual queue architecture proposed for L4S traffic handling in middleboxes. Let us look at some of the key aspects of low latency DOCSIS and how L4S applies in those scenarios.

4.1. Classifying Low Latency Traffic

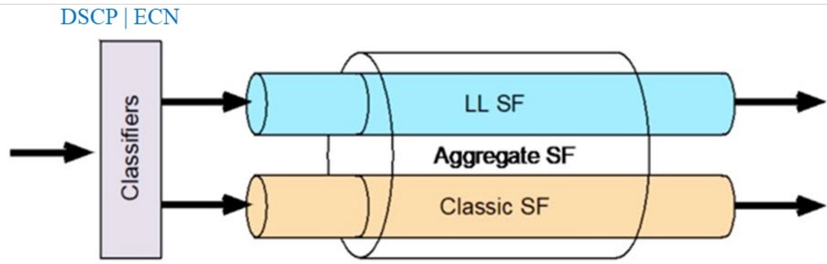


Figure 5 - LLD Classification

Low latency traffic is classified into LL service flow using standard DOCSIS classifiers. While any traffic flow (matched via the supported DOCSIS classification mechanisms) can be directed to low latency handling, the preferred approach is to ensure applications mark themselves appropriately to be directed to LL service flows. This is often achieved by using DSCP marking and [DRAFT NQB-TSVWG] is an active effort to define new DSCP marking for non-queue building traffic.

Alternately, L4S-based applications with ECT (1) marking are expected to be classified into low latency service flows in DOCSIS.

4.2. Queue Protection Function in LLD

Based on the provisioned DOCSIS classifiers, any flow can be directed to the low latency SF. However, overwhelming the LL SF with traffic would result in increased buffer building and the inability to reach the latency goals. To avoid this, LLD defines a queue protection function that tracks which microflow(s) are the cause for increased queue building in the low latency SF and then "sanctions" packets from those microflows to classic SF as shown in Figure 6. A microflow is typically identified as an IP 5-tuple (source port, destination port, source IP, destination IP, and protocol) or similar such constructs.

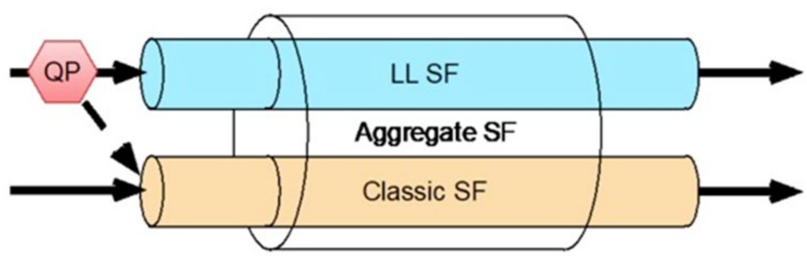


Figure 6 - Queue Protection and Packet Sanctioning in LLD

The queue protection function is a critical component for the smooth functioning of low latency DOCSIS that differentiates it from other differentiated QoS-based queuing systems. The automatic sanctioning of packets to ensure the low latency treatment of NQB traffic is crucial to ensure end-to-end low latency services can be delivered on top of this infrastructure without manual provisioning.

However, high bandwidth traffic that also requires low latency (such as cloud gaming or interactive AR/VR) can result in packet sanctioning and can impact the application latency. This is where transport mechanisms like L4S come into play. By ensuring rapid feedback to senders via ECN marking, L4S

SCTE STANDARDS JOURNAL

streams try to minimize or avoid packet sanctioning to ensure low latency treatment can be met for the service.

4.3. Active Queue Management in LLD

Traditional active queue management algorithms that apply to classic traffic are not sufficient for low latency flows. Low latency AQM requires that packets are marked with CE rather than dropped probabilistically. Also, given that the goal is to provide a CE mark to the senders before queue delay can impact latencies, the low latency AQM needs to act at a much lower timescale (in the order of hundreds of microseconds) and operate on a packet-by-packet basis, unlike some traditional AQM algorithms.

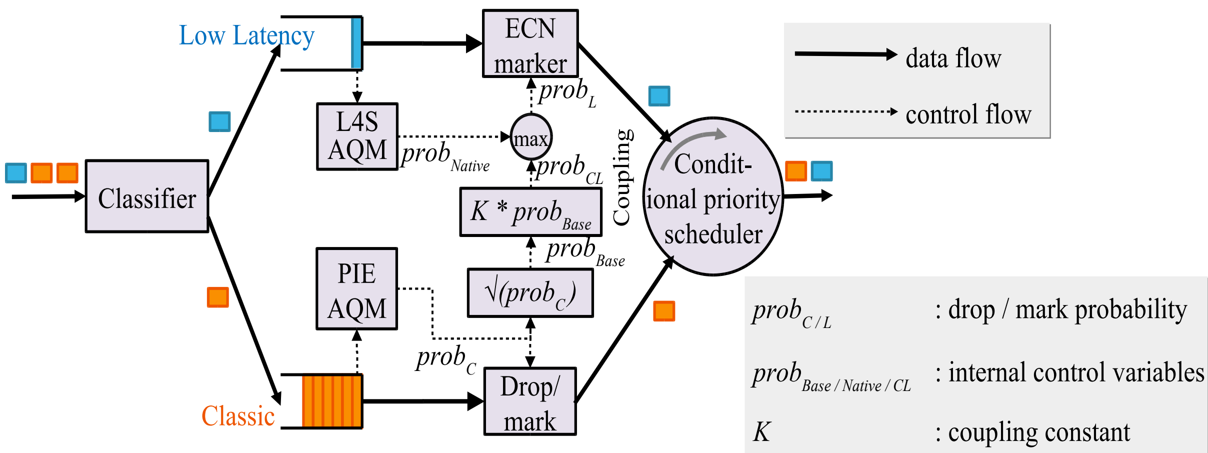


Figure 7 - AQM and Coupling in Low Latency DOCSIS

A special AQM algorithm that supports ECN marking for L4S traffic is defined in low latency DOCSIS and applied to the low latency SF. This is called immediate AQM or IAQM, which acts on a per packet basis and ensures CE marking happens well ahead of queue protection thresholds.

There is also an element of AQM coupling defined in LLD, based on recommendations from L4S architecture. The one-way coupling of drop probability from the classic SF to low latency SF, allows increased marking of low latency traffic in case of possible starvation or heavy queue build-up of classic queues.

As we can observe, the AQM architecture for LLD is derived from and targeted for L4S traffic delivery through the low latency service flow.

4.4. LLD Weighted Scheduler

To ensure low latency traffic is served rapidly, a weighted scheduler (typically weighted round robin, or WRR) is specified in DOCSIS LLD as an inter-SF scheduler within the aggregate service flow. Note that cable access systems may have other fair scheduling algorithms that manage packet scheduling at the overall service group level, in addition to the inter-SF WRR scheduling. The broader goal is to ensure packets that do not cause queue building are served as rapidly as possible, even in heavily congested environments.

The L4S architecture recommends a fair scheduling algorithm between classic and low latency traffic such as a dual queue scheduler. The WRR scheduling defined in LLD is optimal for a dual queue system with the ability to serve NQB flows rapidly in the low latency queue.

Our experiments in the upcoming sections of this document show the impact of tuning this WRR scheduler and how that impacts the throughput behavior of L4S and classic traffic flows.

4.5. Proactive Grant Service

Proactive grant service (PGS) is an upstream-focused optimization defined along with low latency DOCSIS to minimize request-grant delays. Traditionally, DOCSIS upstream transmission is based on the modem requesting a transmission opportunity on data arrival, a CMTS system providing a grant, and the modem using the available grants to complete the transmission. This 3-way handshake often results in increased latencies. PGS allows for a proactive mechanism for CMTS systems to provide a stream of grants to cable modems for low latency traffic. This allows rapid transmission of LL packets arriving at the modem at the next transmission opportunity without waiting for a full grant handshake. Further optimizations are also possible with PGS, including the ability to adapt grant rates based on traffic activity. PGS is a key construct that allows low latency transmission over DOCSIS to be close to the one-millisecond mark.

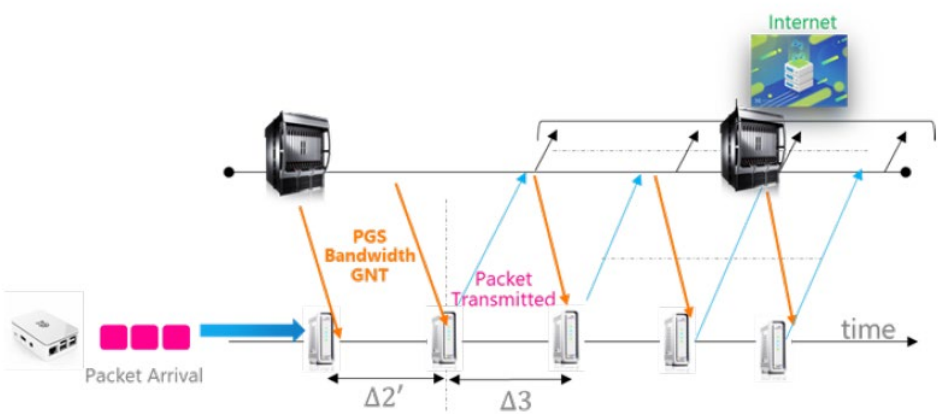


Figure 8 - PGS granting in DOCSIS

In the context of L4S, enabling PGS on low latency upstream service flows allows ultra-low latency services to be provided over DOCSIS networks.

5. Low Latency DOCSIS Measurements

Before looking at L4S experiments and results, understanding DOCSIS LLD metrics would be a useful data point. Our testing of upstream low latency DOCSIS shows that isolating non-queue building traffic from classic traffic via LLD results in marked improvement in latency and jitter, especially with network congestion.

As seen in Figure 9, as the service group utilization increases, the latency of packets through the LL SF is consistently much lower than classic traffic which is subject to queue building. Even as the percentage of low latency traffic continues to increase (to 5% of upstream traffic from 1%), the value proposition of LL traffic isolation to keep low latency is still valid.

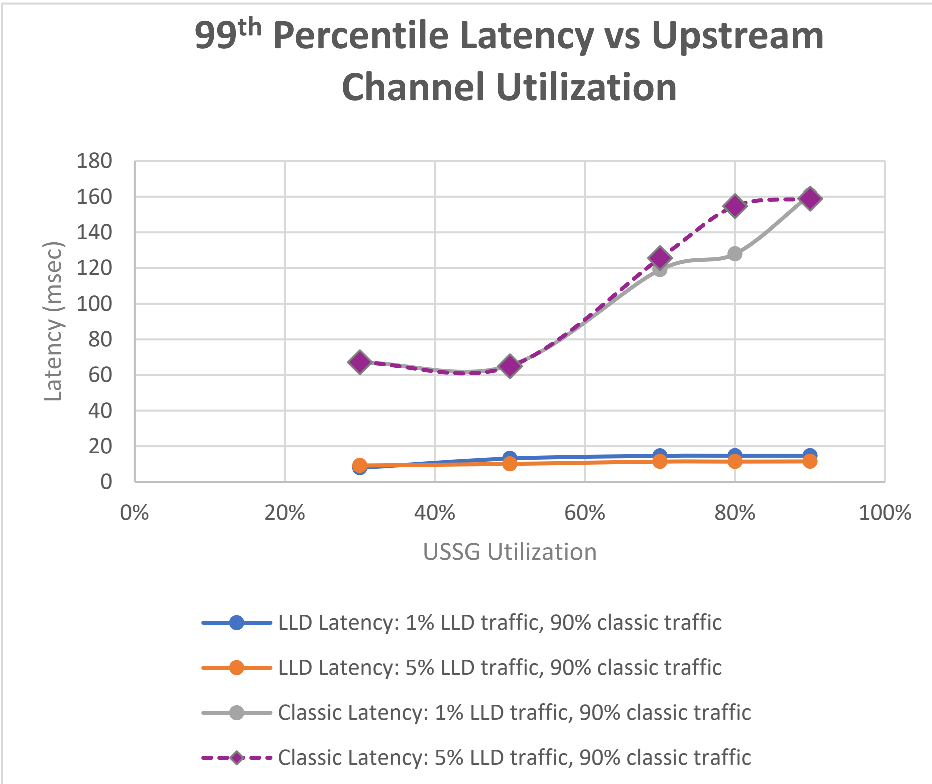


Figure 9 - LLD Upstream Test Results

We also analyzed the behavior of high bandwidth flows through the LL SF to study the impact of queue protection. Figure 10 shows the latency of two microflows through the DOCSIS LLD system while the upstream network and modem are congested like typical "busy-hour" traffic. One microflow referred to as a tracer flow is a lightweight 200kbps stream for the whole duration of the test. The other microflow is the so-called "QP stream," in which the bandwidth of the flow increases from 200kbps to 20Mbps, the upper limit reaching the maximum sustained rate (MSR) of the cable modem. Think of the 200kbps stream as a gaming flow and the QP stream as a VR/AR flow, both being sent to the low latency service flow. For this test, both flows are UDP based while the rest of the network and home traffic are bursty traffic based on Classic TCP.

Based on the traffic and queue building in this specific traffic scenario, we can see how the latency of the QP stream spikes higher when the flow bandwidth is higher than the 6-10Mbps range . This is likely the point at which queue protection kicks in and some of the packets of the high bandwidth flow are being sanctioned to the classic service flow due to queue building.

It should be noted that the inflection point at which this QP sanctioning happens depends on various factors such as overall network load, in-home traffic, configured LLD parameters as well as the bandwidth of the actual LL flows themselves.

L4S is intended to address this specific issue. Instead of the packets of the QP stream being sanctioned, if the sender was able to respond to CE marks aggressively to minimize the chance of queue buildups, the latency of the QP stream could be lower, helping the VR/AR application to achieve end-to-end low latency.

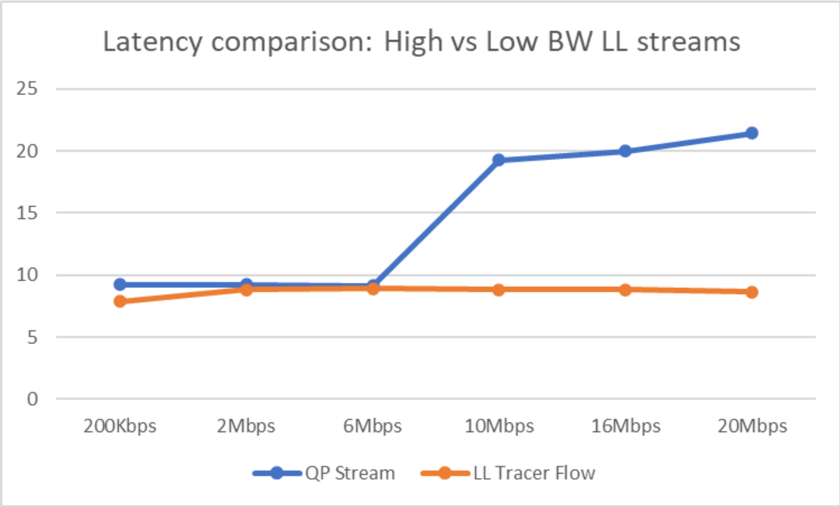


Figure 10 - QP Impact on High Bandwidth LL Stream

6. L4S on DOCSIS Testbed Setup

At the heart of our L4S DOCSIS testbed is an LLD-capable DOCSIS 3.1 cable modem gateway connected to a an Integrated CCAP system running software that is LLD-capable as well.

Since the focus of our testing was on upstream L4S behavior, LLD ASF was only provisioned on the upstream. The traffic sending clients were Linux workstations connected to the home network side of the CM while Linux servers acted as TCP sink connected to the network side interface (NSI) of the CMTS via a 10Gigabit switched network. Client B and Server B are used to carry L4S traffic and have specialized Linux kernels supporting the TCP Prague congestion control algorithm, which allows traffic to be marked as ECT (1) and react to CE signals.

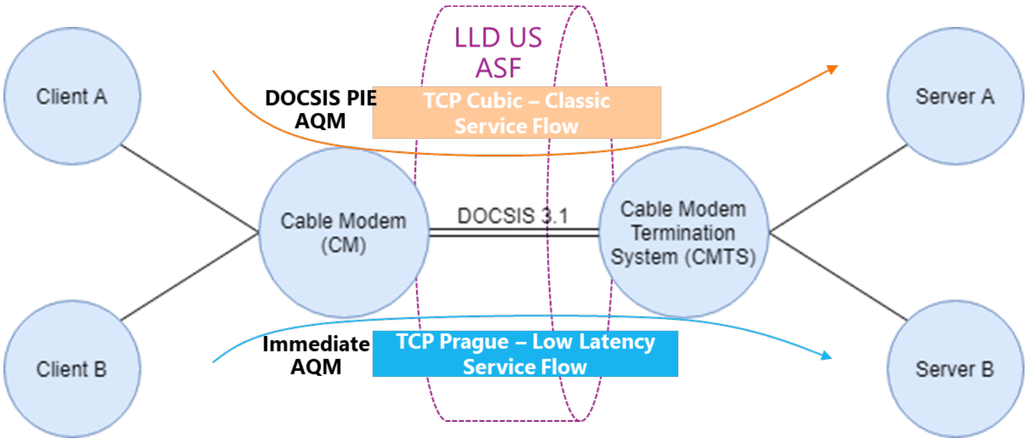


Figure 11 - L4S DOCSIS Experiment Setup

The DOCSIS 3.1 channels were provisioned for an OFDM downstream and OFDMA upstream, to measure the lowest possible latencies of L4S over DOCSIS. Note that LLD technology can be deployed with both SCQAM and OFDM/OFDMA channels (and bonded channel configurations).

7. L4S over DOCSIS: Experiments & Observations

It should be noted that for the experiments below – the terms L4S, TCP Prague, and DCTCP are often used interchangeably. They represent a flow in which packets are marked with ECT (1) signals and packets are marked with CE mark (bits 11) when congestion or queue buildup occurs. In contrast, classic or Cubic flows are traditional TCP streams relying on packet drops for adjusting TCP sending windows.

7.1. L4S baseline tests

For the initial set of L4S testing, the LLD subsystem in DOCSIS was configured to support best effort upstream scheduling for both LL and classic service flows (no PGS was configured for LL SF). This meant the LL traffic still must go through the request-grant acquisition cycle. Typically, this process means the lowest possible DOCSIS roundtrip latency is around 5ms.

However, the default metrics of LLD immediate AQM and queue protection are tuned for a 1ms low latency service. Immediate AQM configuration involves defining a maximum latency threshold and a range function and by default, the maximum threshold is set to 1000 microseconds. Queue protection has two key parameters – the QP latency threshold and QP queuing score threshold. The QP latency threshold should ideally match the IAQM maximum threshold and indicates the latency point at which the queue protection algorithm kicks on. The default value for this metric according to specifications is also 1ms. This is not an ideal setup as it does not account for the DOCSIS request-grant cycle for the upstream.

Figure 12 shows the impact of this default configuration with the L4S (TCP Prague) stream being limited to a throughput under 8Mbps even though the weighted scheduler was provisioned for the LL traffic to use up to 90% of the maximum sustained rate of 50Mbps. This is not an ideal result as the high bandwidth VR/AR stream using L4S will be limited in throughput due to getting excessive congestion signals.

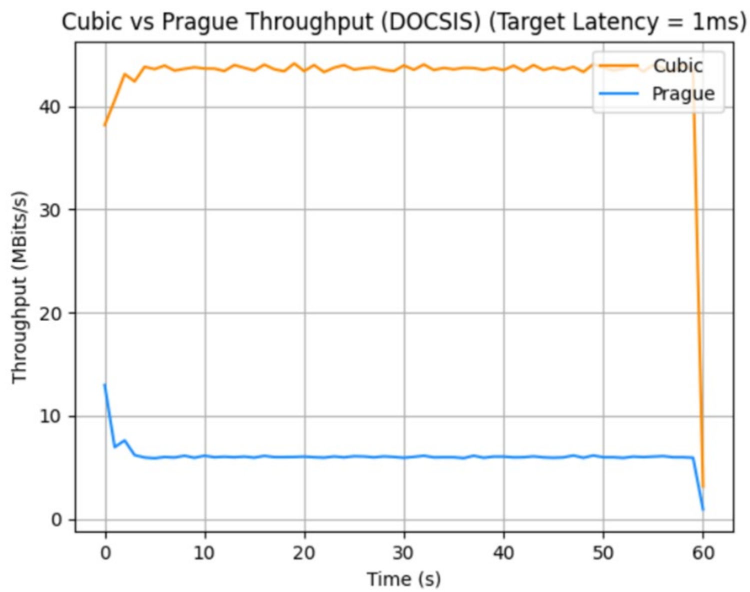


Figure 12 - L4S Poor throughput with 1ms Target Latency

Following this experiment, the configuration of immediate AQM and queue protection was tuned for a target latency of 5ms, which aligns with the DOCSIS request-grant cycle (MAP messaging).

Figure 13 shows the throughput graph with the updated configuration parameters. As can be seen, the L4S/TCP Prague is now able to use up to 45Mbps of the 50Mbps MSR which is the desired behavior.

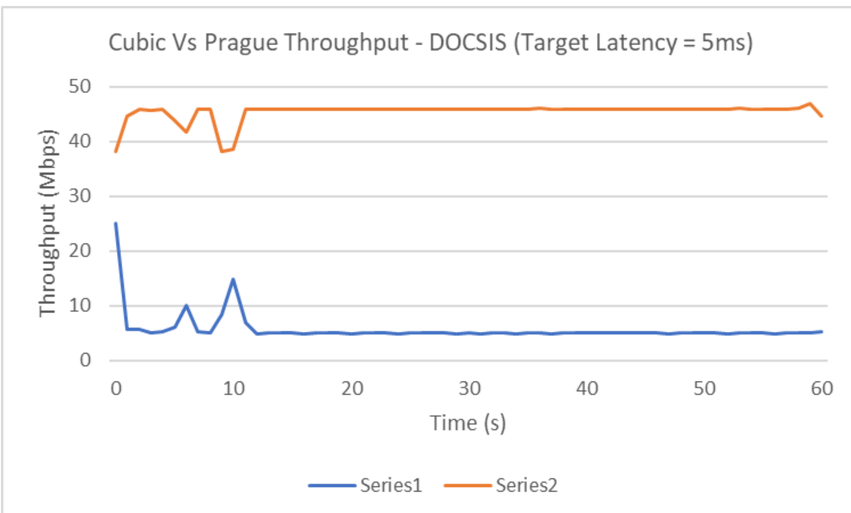


Figure 13 - L4S vs Cubic Baseline Throughput with 5ms Target Latency

We also benchmarked the latency behavior of Cubic and L4S streams for the above scenario. Figure 14 shows that consistently low latency is achieved by the L4S stream.

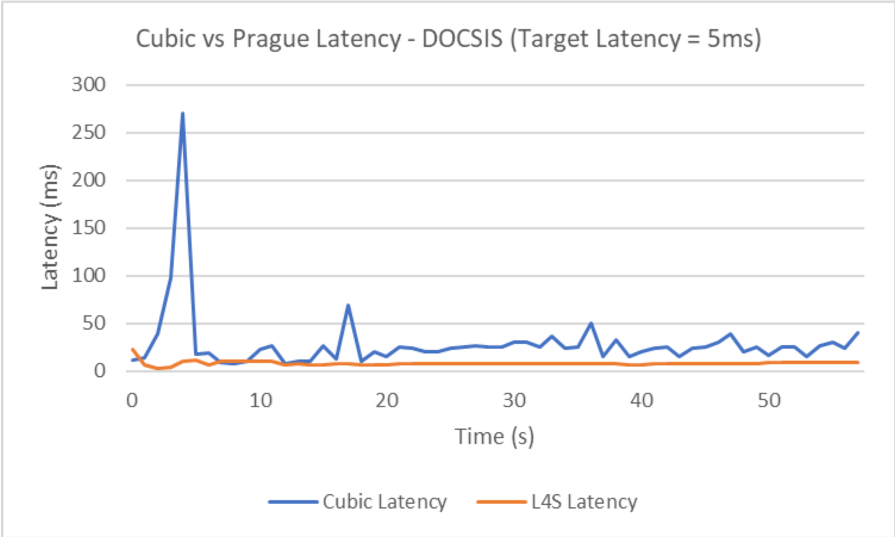


Figure 14 - L4S vs Cubic Baseline Latencies with 5ms Target Latency

The table below shows the details of latency, throughput, and packet loss statistics for the baseline tests.

Table 1 - Baseline L4S vs Cubic metrics

	Latency		**Throughput**		**Packet Loss %**
	Mean	**99th %ile**	**Mean**	**99th %ile**	
TCP Prague (L4S) Stream	6.55ms	7.66ms	44.72Mbps	46.34Mbps	0%
Cubic Stream	28.38ms	161.31ms	5.6Mbps	18.9Mbps	1.27%

It should be noted that there are ongoing standards efforts to improve the IAQM and QP configuration settings in DOCSIS LLD – with the ability to dynamically determine the operating points based on the request-grant cycle.

7.2. AQM Coupling Behavior

AQM coupling is a key construct of the dual queue scheduling system. As a quick background, we should remember that:

1. The classic SF has a DOCSIS proportional integral controller-enhanced (PIE) AQM with a typical target latency of 10ms and drops packets probabilistically based on the algorithm heuristics
2. The low latency SF has an IAQM algorithm that determines probabilistically if a packet must be marked with congestion experienced bits (only for ECN-capable traffic). Additionally, if the queue protection thresholds are exceeded for microflows, packets will be sanctioned to the classic SF where they may be subject to random drops based on classic AQM

Generally, the two AQMs in each of the service flows can act independently. However, the AQM coupling function is employed as a backpressure system – to increase the probability of CE marking for the low latency SF if the classic SF is overwhelmed and dropping packets excessively. The "coupling factor" is a coefficient used in the algorithm for coupling in low latency DOCSIS, with a default value of 2.

Table 2 - Impact of AQM Coupling factor

AQM Coupling factor	L4S Throughput	Cubic Throughput	L4S Mean Latency	Cubic Mean Latency	L4S Pkt Drops	Cubic Pkt Drops
1	44.97	5.63	7.33	27.05	0.00%	1.29%
2	44.72	5.66	6.55	28.38	0.00%	1.27%
3	39.23	10.95	5.22	24.39	0.00%	0.60%
4	34.49	15.53	5.29	19.13	0.00%	0.43%
10	31.76	18.82	5.02	22.78	0.00%	0.60%

Table 2 shows the impact of changing the AQM coupling factor. Increasing the coupling factor allows better "fairness" for cubic traffic at the cost of reduced maximum throughput for L4S streams. The latency impact on changing coupling factor is minimal. The default value of 2 is meaningful if the end goal is the best possible performance for L4S streams. Increasing the coupling factor to 3 or 4 may provide a bit more breathing room to cubic streams – but it does not seem worth changing the coefficient beyond that.

Note that, while these tests were performed at sustained traffic load for the TCP streams, real-world applications are bursty in nature – they try to send at peak rates in short bursts and then have quiet periods before the next burst transmission. And the ability to transmit that burst at the fastest allows the application to have the lowest possible latency.

7.3. Dual Queue Weighted Scheduler

The WRR scheduler in low Llatency DOCSIS ensures non-queue building traffic in the LL service flow is serviced preferably compared to classic traffic. The default metric for the scheduler weight is ~90% in DOCSIS configuration. Combined with the IAQM and queue protection arsenal, this ensures low latency service for latency-sensitive applications while making sure classic traffic is minimally impacted.

Table 3 - Impact of WRR Scheduling Weight

Scheduler Weight	L4S Throughput	Cubic Throughput	L4S Mean Latency	Cubic Mean Latency	L4S Pkt Drops	Cubic Pkt Drops
90%	44.72	5.6	6.55	28.38	0.00%	1.27%
80%	40.15	10.34	7.9	21.06	0.00%	0.54%
70%	34.71	15.48	8.27	21.49	0.00%	0.20%
50%	25.29	24.85	11.37	19.94	0.00%	0.09%

Table 3 shows the throughput, latency, and packet drop metrics for L4S and classic flows with varying scheduler weights. The throughput metrics are self-explanatory – the mean throughput of the two streams fairly reflects the weighted percentage configured. It is interesting to note that even at a 50% scheduling weight, L4S latencies are lower than TCP Cubic while the packet loss stays at 0%. The ability of TCP Prague stack to respond faster to congestion signals helps in keep latencies much lower than waiting to react on packet drops.

7.4. Proactive Granting and L4S traffic

All the above metrics were based on best effort scheduling of low latency traffic. With proactive grant service enabled, over 50% improvement in latencies was observed with L4S traffic with minimal impact on Cubic traffic latency, as shown in Figure 15.

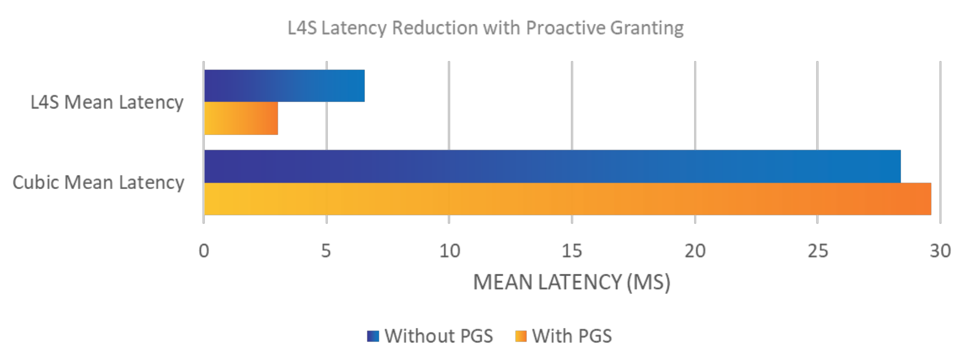


Figure 15 - PGS vs BE Latency Comparison

With fine-tuned configuration settings, DOCSIS round-trip latency for low latency traffic was as low as **2-3ms** in a real-world DOCSIS setup, getting close to the 1ms original target of low latency traffic. Such low access latencies can enable latency-sensitive applications over DOCSIS including Interactive augmented reality apps which demand end-to-end latencies as low as 10ms [GSMA AR/VR].

8. L4S Adoption and Industry Activity

L4S transport is being actively defined, improved, and standardized by the Internet Engineering Task Force and several working documents in the IETF Transport Area Working Group are moving towards RFC Status. Several other standards organizations are working to incorporate L4S and ECN technology as part of the core technology stack, such as the CableLabs low latency DOCSIS effort.

From an industry perspective, application platform providers are actively experimenting with L4S and ECN signaling-based congestion control algorithms. Apple recently announced early support for L4S-based transport in IOS 16 and macOS Ventura [APPLE WWDC-L4S]. Google has incorporated L4S style ECN feedback in TCP BBR version 2 [GOOGLE BBRv2]. Several broadband access vendors have been working on incorporating L4S and ECN support in technologies such as DOCSIS, WiFi, and 5G/cellular systems. L4S and ECN constructs are also being incorporated into several QUIC transport implementations.

The first L4S interop of several major application infrastructure providers and network equipment vendors was achieved during the IETF 114 Hackathon Event in July 2022 [IETF-HACKATHON L4S-TSVWG]. The key focus was on improving the congestion control behavior, achieving fairness between L4S flows against other L4S or Classic TCP flows, and optimizing network behavior. Ongoing interop events and industry activity show a promising future of L4S being deployed at scale on the public Internet.

9. Conclusions

Low latency low loss scalable throughput (L4S) is a new transport technology that can be used for reducing end-to-end network latency for any application. It makes use of Explicit Congestion Notification (ECN) bits in the IP headers and congestion signals are rapidly communicated by the bottleneck points back to the sender. L4S is especially useful for applications that require both high bandwidth and low latency. By rapidly responding to congestion signals and adjusting transmission rates faster, L4S senders can maintain most or all their packets in low latency queues to get the best possible latency, jitter, and packet loss for their application.

Low latency DOCSIS (LLD) defines several primitives such as dual queue scheduler, proactive granting, and other optimizations to reduce latency over the DOCSIS network, especially for non-queue building applications. L4S support is a key component of low latency DOCSIS and our study in this paper is focused on understanding the behavior of L4S and classic streams in an LLD-enabled DOCSIS upstream.

Our study showed the need to understand the behavior of queue protection and immediate AQM algorithms to refine the congestion marking behavior of L4S traffic in DOCSIS. Ongoing specification work is expected to address some of the shortcomings observed in these experiments. Additionally, we studied the impact of some of the key DOCSIS configuration parameters on L4S and Cubic traffic behavior. Our tests also showed that supporting proactive grant service for the DOCSIS upstream allows DOCSIS roundtrip latencies to be under 3ms for L4S traffic.

To conclude, we have observed that L4S streams can take advantage of low latency DOCSIS mechanisms effectively and provide low latency service for applications requiring high throughput. This allows cable broadband networks to be prepared for futuristic applications based on interactivity, virtual reality, augmented reality, and other areas waiting to be explored.

Access network technologies are constantly iterating towards increased bidirectional multi-Gigabit speeds for subscribers – with innovations like XGS-PON, WiFi 6e and 7, 5G eMBB and URLLC services, and DOCSIS 4.0. The innovations in the realms of the metaverse, extended reality (XR), autonomous vehicles and smart cities, and other forward-looking projects often require networks with high throughput along with low latency, jitter, and packet losses. L4S technology along with innovations such as low latency DOCSIS, combined with multi-gigabit speeds are instrumental in preparing our networks for this promising future.

10. Abbreviations and Definitions

AQM	active queue management
AR	augmented reality
ASF	aggregate service flow
BBR	bottleneck bandwidth and roundtrip propagation time (Google developed protocol)
CCAP	Converged Cable Access Protocol
CE	congestion experienced
CMTS	cable modem termination system
DOCSIS	Data Over Cable System Interface Specifications
ECN	Explicit Congestion Notification
eMBB	enhanced mobile croadband
FQ_CoDel	fair queuing controlled delay, an AQM algorithm
IAQM	immediate AQM
IETF	Internet Engineering Task Force
L4S	low latency low loss scalable throughput
LL	low latency
LLD	low latency DOCSIS
MSR	maximum sustained rate
NQB	non-queue building
NSI	network side interface (of the CMTS)
OFDM	orthogonal frequency division multiplexing
OFDMA	orthogonal frequency division multiple access
PGS	proactive grant service
PIE	proportional integral controller-enhanced (an AQM algorithm)
PON	passive optical networks

QoS	quality of service
QUIC	No specific expansion. A transport layer protocol that addresses some of the shortcomings of TCP and typically works in tandem with HTTP/2
QP	queue protection
RFC	request for comments, IETF publication
SCQAM	single carrier – quadrature amplitude modulation
SF	service flow
SG	service group
TCP	Transmission Control Protocol
TOS	type of service
UDP	User Datagram Protocol
URLLC	ultra-reliable low latency communications
VR	virtual reality
WRR	weighted round robin
XGS-PON	A 10G symmetric passive optical network technology
XR	extended reality

11. Bibliography and References

[APPLE WWDC-L4S] "Reducing Network Delays for a more responsive app", V.Goel, https://developer.apple.com/videos/play/wwdc2022/10078/

[DOCSIS MULPI] "DOCSIS 3.1 MAC and Upper Layer Protocols Interface Specification", CableLabs

[GOOGLE BBRv2] "BBRv2: A Model-based Congestion Control", https://datatracker.ietf.org/meeting/105/materials/slides-105-iccrg-bbr-v2-a-model-based-congestion-control-00

[GSMA AR/VR] "Cloud AR/VR Whitepaper", GSMA Future Networks

[IETF-DRAFT DUALQAQM-TSVWG] "Dual Queue Coupled AQM for Low Latency Low Loss Scalable Throughput (L4S)", https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-aqm-dualq-coupled-25

[IETF-DRAFT L4SARCH-TSVWG] "Low Latency Low Loss Scalable Throughput Internet Service: Architecture", https://datatracker.ietf.org/doc/draft-ietf-tsvwg-l4s-arch/

[IETF-DRAFT L4SECNID-TSVWG] "Explicit Congestion Notification (ECN) Protocol for Very Low Queuing Delay (L4S)", https://datatracker.ietf.org/doc/draft-ietf-tsvwg-ecn-l4s-id/

[IETF-DRAFT L4SOPS-TSVWG] "Operational Guidance for Deployment of L4S in the Internet", https://datatracker.ietf.org/doc/draft-ietf-tsvwg-l4sops/

[IETF-DRAFT NQB-TSVWG] "A Non-Queue Building Per-Hop Behavior for Differentiated Services", https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-nqb-11

[IETF-HACKATHON L4S-TSVWG] "First L4S Interop Event @IETF Hackathon", https://datatracker.ietf.org/meeting/114/materials/slides-114-tsvwg-update-on-l4s-work-in-ietf-114-hackathon-00.pdf

[MATHUR-2020] "Low Latency DOCSIS – Concepts and Experiments", T.Mathur, R. Ranganathan, B.Zhang, G.Gohman, SCTE 2020

[RFC 3168] The addition of Explicit Congestion Notification to IP https://datatracker.ietf.org/doc/html/rfc3168

SCTE STANDARDS JOURNAL

Optimizing Wi-Fi Channel Selection in a Dense Neighborhood

A Technical Paper prepared for SCTE by

Yonatan Vaizman
Lead Researcher, Machine Learning
Applied AI & Discovery, Comcast
1325 G Street NW, Washington, DC 20005
202-524-5077
Yonatan_Vaizman@comcast.com

Hongcheng Wang
Distinguished Engineer, Machine Learning
Applied AI & Discovery, Comcast
1325 G Street NW, Washington, DC 20005
332-301-5055
Hongcheng_Wang@comcast.com

Table of Contents

Title **Page Number**

Table of Contents 62
1. Introduction 63
2. Wi-Fi Pain Metric 63
3. Optimization Problem and Solvers 65
   3.1. MIQP Problem Solver 66
   3.2. Neural Network Gradient Descent 66
4. Preliminary Experiments 68
   4.1. Estimating Potential Pain 68
   4.2. Optimization Details 70
   4.3. Results 70
5. Conclusion 70
   5.1. Future Directions 71
6. Acknowledgements 71

Abbreviations 71

Bibliography & References 72

List of Figures

Title **Page Number**

Figure 1 - Channel Allocation for a Dense Area 65

Figure 2 - Neural Network Approach with Gradient Decent (GD) and Back Propagation 67

Figure 3 - Estimating the Co-Usage, Sensing, and Potential Pain Matrices 69

List of Tables

Title **Page Number**

Table 1 - Experimental Results 70

1. Introduction

In dense neighborhoods, there are often dozens of homes in close proximity. This can either be a tight city-block with many single-family homes (SFHs), or a multiple dwelling units (MDU) complex (such as a big apartment building or condominium). Each home in such a neighborhood (either a SFH or a single unit in a MDU complex) has its own Wi-Fi access point (AP). Because there are few (typically 2 or 3) non-overlapping radio channels for Wi-Fi, neighboring homes may find themselves sharing a channel and competing over airtime, which may cause bad experience of slow internet (long latency, buffering while streaming movies, *etc.*). Existing APs sometimes have smart channel selection features – typically scanning the air to select the least occupied channel. However, because they work independently (the APs do not coordinate), this can cause a cascade of neighboring APs constantly switching channels, which is disruptive to the connectivity of the homes. Wi-Fi optimization over all the APs in a dense neighborhood is highly desired to provide the best user experience.

We present a method for Wi-Fi channel selection in a *centralized* way for all the APs in a dense neighborhood. We describe how to use recent observations to estimate the potential-pain matrix: for each pair of APs, how much Wi-Fi-pain would they cause each other if they were on the same channel. We formulate an optimization problem – finding a channel allocation (which channel each home should use) that minimizes the total Wi-Fi-pain in the neighborhood. We design an optimization algorithm that uses gradient descent over a neural network to solve the optimization problem. We describe initial results from offline experiments comparing our optimization solver to an off-the-shelf mixed-integer-programming solver. In our experiments we show that the off-the-shelf solver manages to find a better (lower total pain) solution on the train data (from the recent days), but our neural-network solver *generalizes* better – it finds a solution that achieves lower total pain for the test data ("tomorrow").

We discussed this work in the 2022 Fall Technical Forum as part of SCTE Cable-Tec Expo®.

2. Wi-Fi Pain Metric

To measure the pain caused to the users in a dense Wi-Fi space, we define a new Wi-Fi pain metric. The main cause for Wi-Fi density pain is when a home's neighbors are using the same radio channel and occupying much of its airtime: when my home's AP senses high interference because others are using the same channel, my home's devices (including my AP) will have to wait longer times before they can send their packets over the radio channel, and this will cause slowness and subpar user experiences.

However, if my home barely has internet traffic during the night, while my neighbors use the same Wi-Fi channel heavily at the same time, that interference doesn't cause me any pain. The pain comes when my neighbors use the channel heavily *while* my home tries to use the same channel.

In addition, my home may have a lot of internet traffic at the same time as another home in my apartment building, but because there are five floors separating the two homes, our Wi-Fi signals never interfere with each other (the homes cannot "sense" each other – we will define this more formally later).

To simplify, we notice that in a dense neighborhood, homes cause each other Wi-Fi pain when three conditions are met: the homes can sense each other, they tend to have a lot of internet traffic at the same times, and they use the same radio channel. The first two are regarded as given conditions of the neighborhood (we can measure or estimate them, but we cannot control them) and the third is the aspect that we can control – which channel does each home use. We treat these three components as independent. Let's now formalize the overall pain mathematically with these three components, for a neighborhood with n homes and n_c Wi-Fi channels:

- The **(binary) sensing** matrix, $S^b \in \{0,1\}^{n\times n}$. ${S^b}_{i,j}$ is 1 *if and only if* home i can sense (and be interfered by) home j.
- The **co-usage matrix** $U \in \mathbb{R}_+^{n\times n}$. This describes how much homes tend to have internet traffic at the same time. Notice, it doesn't matter which channel each home is using, and it doesn't matter if the homes can sense each other. This component only cares about the behavior patterns of the homes' residents and devices (specifically, the internet-activity patterns).
- The **channel allocation matrix**: $C \in \{0,1\}^{n\times n_c}$. For each home (row), which channel is assigned to it – exactly one channel (out of the n_c options) has a value of 1. Typically, n_c is 2 or 3.

The pain that home j causes to home i depends on the three conditions we mentioned – this is expressed with multiplication:

$\sum_{c=1}^{n_c} {S^b}_{i,j} U_{i,j} C_{i,c} C_{j,c}$.

Notice, that we use matrix C twice in the formula and inside a sum over the possible channels (c) – this is to capture if the two homes are using the same channel: if the two homes are not on the same channel, the whole sum will be 0, but if they are on the same channel, the sum will have a single non-zero element ${S^b}_{i,j} U_{i,j}$. Similarly, if the two homes don't even sense each other (${S^b}_{i,j} = 0$), the whole sum will be 0 (even if they are using the same channel) – this can describe two homes that are physically far away from each other in the neighborhood, or have many walls between them, so the radio signal doesn't travel from one to the other. We assume additivity: the pain that home i senses from the neighborhood is the sum of the pain that it senses from all the neighborhood's homes:

$pain_i = \sum_{c=1}^{n_c} \sum_{j=1}^{n} {S^b}_{i,j} U_{i,j} C_{i,c} C_{j,c}$.

To simplify the formula, we combine the two components that we cannot control and define the **potential-pain matrix** $P = S^b \circ U$ (elementwise multiplication). $P_{i,j} = {S^b}_{i,j} U_{i,j}$ describes the pain that home j would add to home i if they were using the same channel. The total pain in the neighborhood is a sum over the homes:

$$pain^{total} = \sum_{c=1}^{n_c} \sum_{i,j=1}^{n} {S^b}_{i,j} U_{i,j} C_{i,c} C_{j,c} = \sum_{c=1}^{n_c} \sum_{i,j=1}^{n} P_{i,j} C_{i,c} C_{j,c}$$

And we can express it in matrix form:

$$pain^{total} = \sum_{c=1}^{n_c} [C^T PC]_{c,c} = Tr(C^T PC)$$

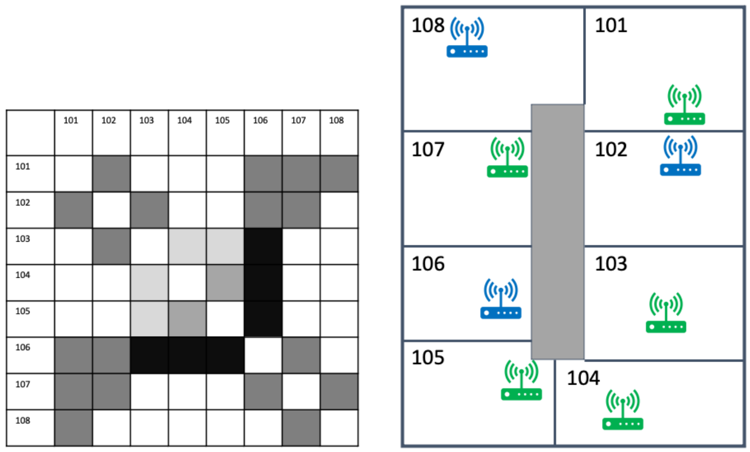


Figure 1 - Channel Allocation for a Dense Area

Figure 1 illustrates part of a made-up dense neighborhood (right image) – a floor plan with 8 apartments in an apartment building, and the potential-pain matrix for the 8 homes (left image), where darker shades of gray represent higher potential-pain value. The floor plan in the figure has two colors to the APs in the homes, representing a possible channel-allocation to two channels (blue and green).

Homes 101 and 104 are far away from each other (see the floor plan), so their APs never sense each other – this explains why they have a blank (0) value in the matrix – they have 0 potential to cause each other pain. This also explains why a smart channel allocation may allocate the same channel (green) to these two homes.

Home 106 represents a heavy internet user (most of the day has a lot of traffic), so it has the potential to cause much pain (darker shade in the matrix) to the homes that can sense it and typically have internet traffic at the same times (103, 104, 105). Homes 101 and 102 can sense home 106, but they may have internet traffic at different times of the day than home 106, so they have lower potential pain from 106 (medium gray shade). It makes sense to put home 106 on the blue channel and isolate it from homes 103, 104, and 105 (allocated the green channel).

3. Optimization Problem and Solvers

We can now define the main optimization problem as follows:

$$C^* = arg \min_{C \in \{0,1\}^{n \times n_c}} Tr(C^T P C)$$
$$s.t.$$
$$\forall i \in \{1 \dots n\}: \sum_{c=1}^{n_c} C_{i,c} = 1$$

This problem assumes we know (or estimate from recent data) the potential-pain matrix P – it is the conditions of the neighborhood, the potential of homes to cause Wi-Fi pain to one another. The task of the optimization is to select a good combination of per-home channels, to minimize the overall pain that the homes cause each other. One of the reasons for this *centralized* channel selection approach is to avoid too many channel changes – frequent changes can be disruptive to the users' connectivity experience. So, a typical use would be to solve this optimization problem, set the selected channels to all the neighborhood's APs, and keep the channels fixed for a while (e.g., a whole day, a whole week).

3.1. MIQP Problem Solver

We note that our optimization problem is a mixed-integer quadratic programming (MIQP) problem: the search parameter C appears in the objective function (the formula for total pain) in a quadratic form, and its values are constrained to be integers. This is a non-convex problem, and we don't have an algorithm that can guarantee finding the global optimum (the very best combination of per-home channels) in reasonable time.

There are commercially available solvers, like Gurobi (Gurobi Optimization, 2022), that use a branch-and-bound approach to solve mixed integer programming problems (including the quadratic type). These methods iteratively try to rule out parts of the parameter-space and narrow down where we can find the global optimum, as well as narrow down the gap between lower and upper bounds for the optimal objective value. These tools often manage to reach the global optimum and they employ various heuristics to try to speed up the process.

3.2. Neural Network Gradient Descent

We propose an alternative method to solve the optimization problem. We construct a neural network model to calculate a soft-approximation of the neighborhood's total pain, given any combination of channel allocation, and use gradient descent with back-propagation to change the underlying parameters until the pain reduces to a local minimum. The model is illustrated in Figure 2.

The model's parameters are represented as a matrix $W \in \mathbb{R}^{n \times n_c}$. The input to the model is a dummy scalar variable $\beta \in \mathbb{R}_+$. Using W and β, the model calculates a "soft" version of channel allocation $C^{\beta,W} \in [0,1]^{n \times n_c}$ by using the softmax operation on each row of βW:

$$C_{i,c}^{\beta,W} = \frac{e^{\beta W_{i,c}}}{\sum_{d=1}^{n_c} e^{\beta W_{i,d}}} .$$

The resulting matrix $C^{\beta,W}$ has each row (for home i) describing a probability distribution over the n_c optional channels. This is not a valid channel allocation (in practice each AP only uses a single channel at a time), but this is a soft approximation of a valid channel allocation.

The model then incorporates the potential pain matrix P as a fixed given input and uses $C^{\beta,W}$ to calculate a soft approximation of the total pain:

$$pain^{\beta,W} = Tr\left(C^{\beta,W^T} P C^{\beta,W}\right).$$

Notice, that the input variable β controls the order of the approximation: with a small value, like $\beta = 0.1$ the soft channel allocations in $C^{\beta,W}$ will be closer to a uniform distribution over the n_c channels. With a higher value, like $\beta = 100$, the soft channel allocations better approximate a valid channel allocation – where for each home only a single channel gets a value close to 1 and the other channels get a value close to 0.

To solve the optimization problem, we start by randomly initializing the parameters W (e.g., using an *i.i.d.* standard normal distribution), and then use gradient descent (with back-propagation) to reduce the approximated total pain $pain^{\beta,W}$. In addition, we start by using a small value of β as input, and slowly increase it. This helps the algorithm first find a good global area and only later fine tune the parameters to a local minimum. After this procedure converges to a local minimum, and the parameters are tuned to values W^{end}, we can get the solution (the chosen channel allocation) by looking at the approximated channel allocations (for large β) and thresholding their values:

$$C_{i,c}^{end} = 1\left[C_{i,c}^{1000,W^{end}} > 0.5\right].$$

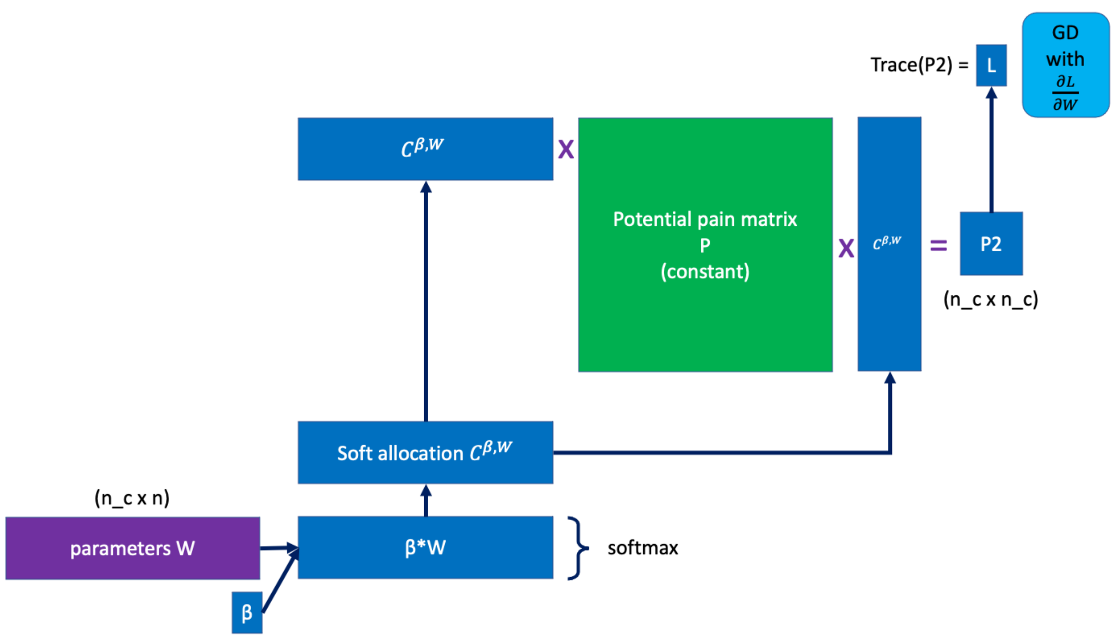


Figure 2 - Neural Network Approach with Gradient Decent (GD) and Back Propagation

While this gradient descent approach does not presume to find a better (lower) optimum than off-the-shelf solvers, we want to highlight a few advantages it has:

- This approach doesn't assume that the potential pain matrix P is symmetric, while other methods may rely on convex relaxations of the optimization problem, requiring them to have a symmetric (and positive-semi-definite) matrix for the quadratic form.

- This approach can be modified to solve a different optimization problem that tries to minimize the *worst-home-pain* instead of the total, or *average-home-pain*. By making slight changes to the neural network, it can approximate the pain of the worst suffering home, and the optimization will try to minimize that value.

- It is simple to add modifications that are common in training neural networks for typical supervised machine learning. We can use momentum when updating the parameters, for faster convergence. We can add parameter regularization (like the L_2 norm penalty – $\lambda\|W\|_2^2$) to the loss function, to avoid "overoptimizing" to the estimated potential-pain matrix.

- This approach runs efficiently, quickly reaching a local minimum.

- This approach does not overemphasize getting to the global minimum. We want to generalize to near-future data, so we should avoid overfitting to the most recent days' data.

4. Preliminary Experiments

During summer 2021, we conducted a few offline experiments with data from a big apartment building. We had data from 66 homes in the building, so we treated them as "the neighborhood's homes" for the experiment. We tried various combinations of different aspects, and we share here some of our preliminary experiments and results. In these experiments, we simulated running the optimization on a reference date, to select the channel allocation for the following day. We collected data from the homes in the neighborhood from the recent days up to (and including) the reference day (the train days), calculated the potential pain matrix, and solved the channel allocation problem. We did a similar calculation to get the potential pain matrix for the day following the reference day (the test day). We evaluated the total pain on both the train days and the test day, given the chosen channel allocation, keeping in mind that the real goal is to improve (minimize) the pain on the test day.

4.1. Estimating Potential Pain

We estimated the two components of the potential pain separately: the (binary) sensing matrix S^b and the internet co-usage matrix U. Figure 3 illustrates this process: the colors in the matrices represent the cell values, ranging from 0 (dark blue) to high values (bright yellow). Each matrix has a different range (see the color-bar to the right of each image).

The co-usage can be defined as some version of multiplying two home's internet-traffic time-series ($u_{t,i}$ represents home i's usage at time t). In this paper we use $U_{i,j} = log\left(1 + \sum_t u_{t,i}u_{t,j}\right)$, but we can have many variations: sum each home's time-series first and then multiply, use a different non-linearity than logarithmic, apply non-linearity on $u_{t,i}$ alone to produce a non-symmetric version, *etc*. To estimate the co-usage matrix U, we used periodic measurements that each AP takes every 15 minutes. Specifically, we used a measurement of percentage of airtime that the AP occupied the channel to *transmit* data to the home's devices (the "download" direction, assumed to occupy the majority of airtime in a typical home). We smoothed the quarter-hourly measurements to hourly quantities. We experimented with both measurements from whole-days (all hours of the day) and evening-time (only using measurements from 7pm-10pm local time), but here we focus our results on evening-time. For estimation with the recent n_d days, this results in a time-series (vector) of $3n_d$ hourly values for each home. We then calculated the cross-correlation between homes (the dot product of two homes' time-series) and took the $\log(1 + x)$ of these values.

The top row of Figure 3 illustrates the process of estimating the co-usage matrix: starting with a Wi-Fi usage time-series for each home (top left). The image shows 10 homes and airtime-percentage values from 96 time points. This narrow matrix is multiplied by its transpose to produce the usage correlation matrix (for each pair of homes the value is the dot product of their two time-series). These correlation values can be extremely large (notice the color-bar reaching values of 200k), so we then apply logarithmic compression to form the co-usage matrix U.

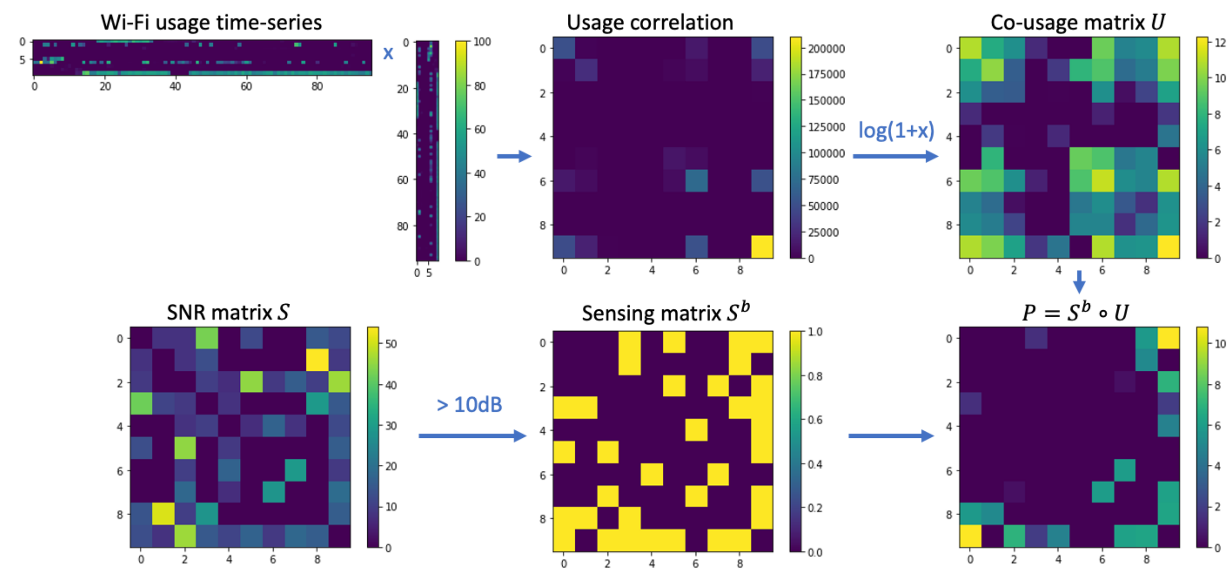


Figure 3 - Estimating the Co-Usage, Sensing, and Potential Pain Matrices

For estimating the sensing matrix S^b, we used radio-scan reports from the APs in the neighborhood: each AP performs a scan multiple times a day to look for Wi-Fi beacons in the air. The AP records the media access control (MAC) address of every other AP that it senses, and the signal to noise ratio (SNR) of the sensed beacon. We mapped sensed Wi-Fi MAC addresses to the familiar APs that are part of the neighborhood. The scans reported additional sensed entities that came from external APs (which we don't know and cannot control). For each pair of homes $\langle i, j \rangle$ in the neighborhood, we averaged the SNR values (over a period, like a week) of how strongly home i's AP senses home j's AP. We can call these variables the SNR matrix S (typically having non-negative real values), illustrated in Figure 3 bottom left image for 10 homes. In our experiments, we chose to symmetrize the sensing matrix: $S \leftarrow 0.5(S + S^T)$. We applied a threshold of 10dB to produce the binary sensing matrix S^b (Figure 3, bottom middle image). Notice that since an AP never sensed itself in the radio scans, we naturally get zeros in the diagonal. This fits our formulation, because we wish to only model the pain that homes cause other homes, not themselves.

We multiplied (elementwise) these two estimated matrices U and S^b to form the potential pain matrix P (bottom right image in Figure 3).

For the test day, we calculated the co-usage matrix U from the test day's usage measurements. However, we used the same SNR matrix S as we did for the train days. This is because we didn't have sufficient scan measurements from every day, and because we assumed that who can sense whom stayed stationary over a longer time (~a month).

4.2. Optimization Details

We used the Gurobi package (Gurobi Optimization, 2022) as a MIQP solver. For our neural network algorithm, we implemented the network using TensorFlow (Martín Abadi, 2015) and Keras (Chollet, 2015). Every update step had just a single example input into the network. We increased the value of the input variable β in phases (running 6,400 update steps in each phase) with values: 1, 10, 100, 1000. We used ADAM optimizer with learning rate 0.001.

4.3. Results

Table 1 - Experimental Results

	Train days	Algorithm	Total pain – per train day	Total pain – test day
1.	1 (Aug 24)	Gurobi	58.2	194.0
2.	1 (Aug 24)	Neural Network	58.2	184.9
3.	4 (Aug 21-24)	Gurobi	64.5	166.3
4.	4 (Aug 21-24)	Neural Network	69.9	143.8

We show in Table 1 results from a few of our offline experiments with a single neighborhood. These were all done with train days up to (and including) August 24th and testing on usage data from August 25th. In these experiments, we used usage (and scan information) from the 2.4GHz frequency and we simulated solving the channel allocation for $n_c = 2$ channels. Rows 1-2 show experiments where there was only a single training day, compared to 4 training days in rows 3-4 (the table reports the average total pain per train day). The results show that when training with data from more days, we could achieve a worse (higher) total pain on the train data, but a better (lower) total pain on the test day, which is what we want to achieve. As expected, our neural network solver did not beat Gurobi's solution on the train days. However, the neural network solver's solution generalized better to the test day – it achieved a lower pain than Gurobi's solution (in both the 1-train-day and 4-train-days scenarios).

5. Conclusion

We have discussed the problem of Wi-Fi airtime competition in a dense neighborhood and the need for a centralized channel selection solution. We defined a Wi-Fi pain objective, based on the co-occurrence of close neighbors having a lot of internet traffic at the same time on the same radio channel. We formulated the pain such that all the relevant information is captured in a single square matrix P, indicating for each pair of homes how much pain would one add to the other if they were using the same channel. We formulated an optimization problem and offered two alternative solvers for it: an off-the-shelf MIQP problem solver and a tailored neural network solver. We conducted preliminary offline experiments with data from a real neighborhood and demonstrated how we can achieve better generalization (lower pain for "tomorrow") with more training days and by using our neural network solver.

5.1. Future Directions

There are still many more directions to research. We can explore various flavors of defining Wi-Fi pain: there can be non-symmetric definitions of potential-pain, for example, when one AP tends to transmit with higher power than a neighboring AP. We can incorporate external sources of interference into the pain model, for example it is possible that the lower floors of an apartment building consistently experience interference in a particular radio channel from a nearby store. When estimating the potential-pain based on the recent month, we may want to give different weight to different days of the week, to get a better estimation of what is about to happen tomorrow. Different neighborhoods may require different approaches – a suburban condominium with long term residents may be a good candidate for estimating the potential-pain based on a whole month, while a big apartment building in a busy city block may have faster turnaround of residents and may require estimation based on the most recent few days.

The optimization algorithm can have various adjustments. We can add regularization on the parameters W, or even constraints on the output values of some of the nodes in the neural network. The schedule of changing β may influence the outcome. An interesting direction is minimizing the worst-home pain and seeing how it influences the average-home pain. To explore this direction, we need to adjust the neural network: instead of calculating the whole quadratic form of $C^{\beta,W}$, the network will first calculate the approximated total pain for each home individually, and then apply a soft approximation of the max operation, to pick the most suffering home in the neighborhood.

We will conduct more offline experiments with many more neighborhoods. Additionally, actual trials will reveal more reliably how helpful is centralized channel selection and A/B tests can help demonstrate which methods are better. We can use a contextual-bandit approach to cleverly select the appropriate "flavor" of Wi-Fi pain for each neighborhood (*e.g.,* how many days to use when estimating the potential-pain). In actual channel-selection experiments, we can more directly measure the sensed interference that every AP experiences from its environment. More importantly, we'll have to assess the effect on the residents' subjective experience of slow internet and Wi-Fi pain.

6. Acknowledgements

We thank Zhehao (Kenny) Zhang for running the offline experiments, and Shelley Leung for processing telemetry data. We thank Obi Asinugo, Zekun (Katherine) Yang, Ali Mohammadi, Teddy ElRashidy, and Colleen Szymanik for great discussions and guidance about this interesting Wi-Fi problem.

Abbreviations

AP	access point
DFS	dynamic frequency selection
Hz	Hertz
SCTE	Society of Cable Telecommunications Engineers
MIQP	mixed integer quadratic programming
MAC	media access control
SFH	single-family home
MDU	multiple dwelling unit

SNR	signal to noise ratio

Bibliography & References

Chollet, F. a. (2015). *Keras.* Retrieved from https://keras.io
Gurobi Optimization, L. (2022). *Gurobi Optimizer Reference Manual*. Retrieved from https://www.gurobi.com
Martín Abadi, A. A. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems.* Retrieved from https://www.tensorflow.org/

SCTE STANDARDS JOURNAL

MLOps and ML Platforms: An Overview

Best Practices for AI/ML in Production

A Technical Paper prepared for SCTE by

Nicholas Pinckernell
Distinguished Engineer – ML, Comcast, SCTE Member
183 Inverness Dr W
Englewood, CO
nicholas_pinckernell@comcast.com

Jan Neumann
Vice President - ML, Comcast, SCTE Member
1325 G St NW
Washington, DC 20005
jan_neumann@cable.comcast.com
(202) 790-1036

Table of Contents

Title **Page Number**

Table of Contents 74
1. Introduction 76
2. DevOps to MLOps 76
3. The ML Life Cycle 77
   3.1. Features 80
   3.2. Models 81
      3.2.1. Model Improvement 82
   3.3. Model Accuracy 83
      3.3.1. Supervised 83
      3.3.2. Weakly Supervised 84
      3.3.3. Model Bias 85
4. ML Platform Requirements 85
   4.1. General Consideration 85
   4.2. Requirements for different Personas 85
      4.2.1. Business or Product Owner 85
      4.2.1. Data Scientist or Researcher 86
      4.2.1. Data or ML Engineer 86
5. Enabling MLOps with ML Platforms 87
   5.1. Common Functions of ML Platforms 88
6. Example ML Platforms and Systems 88
   6.1. AWS SageMaker 89
   6.2. TensorFlow TFX 89
   6.3. Azure Machine Learning 89
   6.4. Databricks 89
   6.5. H2O 89
   6.6. Kubeflow 89
7. Conclusions 90
8. Abbreviations and Definitions 90
   8.1. Abbreviations 90
   8.2. Definitions 90

References 91

List of Figures

Title **Page Number**

Figure 1- Value of Ideas, Code and Production Code 76

Figure 2 - The DevOps Life Cycle 77

Figure 3 - The AI/ML Life Cycle 79

Figure 4 - SDLC and ML Life Cycle Commonalities 80

Figure 5 - Kitten in Grass (from https://unsplash.com/photos/RCfi7vgJjUY) 81

Figure 6 – DevOps vs MLOps Pipeline 82

Figure 7 – AWS SageMaker Ground Truth Framework 83

Figure 8 – Snorkel Overview 84
Figure 9 - AI/ML Platform Components 87

SCTE STANDARDS JOURNAL

1. Introduction

Machine Learning (ML) and Artificial Intelligence (AI) are part of every company's strategic plan and with the opportunities they offer to augment, optimize, and automate tasks that require "intelligence" in the general sense, many organizations are planning to increase the number of AI/ML solutions used.

Machine Learning is referred to by some (e.g., Tesla's chief scientist Andrej Karpathy (Karpathy, n.d.)) as Software 2.0, in the sense that it augments traditional software by including components that were created by ML training. ML training in turn can be interpreted as an automated search for the best program to achieve the desired objective, instead manual creation of a program using heuristics.

The development and deployment of AI/ML solutions is more complex than traditional software development due to the need to consider both the data and model pipelines, in addition to the software development pipelines. In this paper, we will collect the best practices for the development, deployment and operation of AI/ML solutions.

2. DevOps to MLOps

While most engineers are aware of how to run a successful software project (e.g., you need QA, monitoring, and many other components to support your project and continue to deliver features), the same goes for machine learning models and processes. However, before specific technologies can be assessed it is important to remember that "Ideas, in and of themselves, have no value. Code, in and of itself, has no value. The only thing that has value is code that is running in production. Only then have you created value for users and had the opportunity to impact your business." as succinctly stated in article (Edge Computing at Chick-fil-A, n.d.).

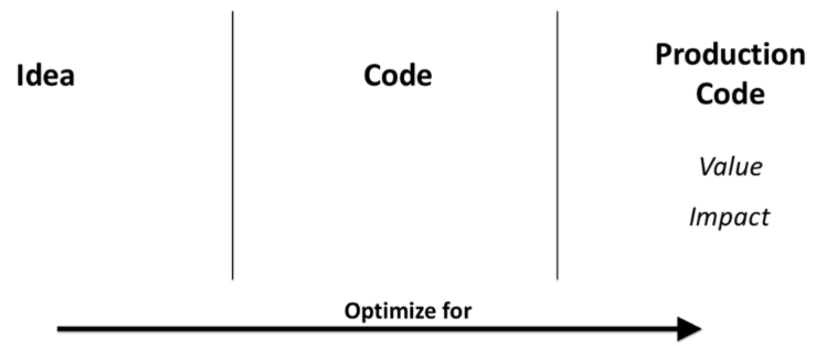


Figure 1- Value of Ideas, Code and Production Code

Additionally, it has been shown by the book Accelerate (Jez Humble, 2018) that the specific technology does not correlate to success. What does correlate to success is the core of DevOps – the ability to turn ideas into code and get code into production rapidly.

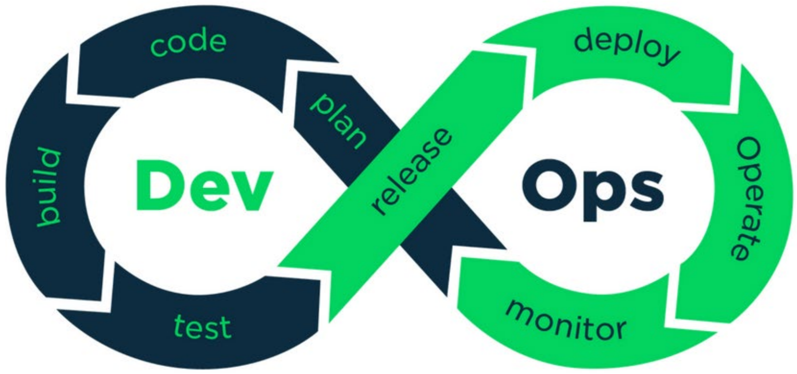


Figure 2 - The DevOps Life Cycle

Just as the DevOps life cycle (see Figure 2) has revolutionized the processes and tools to support software development, the same principles can be applied to the development of machine learning driven solutions. According to the Wikipedia definition of DevOps (DevOps, Wikipedia, n.d.):

> "DevOps is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality …"

Similarly, MLOps encapsulates the combination of data science, data engineering and software engineering to solve business problems using automatically applied insights and actions based on data. If we update the core goal of DevOps for MLOps we get to the ability to turn ideas into data pipelines, machine learning models and integration code, and all three of those into a solution in production quickly and efficiently.

Many AI/ML solutions never make it into production due to the complexity of the process, which is often still very manual and thus error prone. MLOps provides the tools and processes to glue the different steps in the ML lifecycle together in robust and repeatable manner, thus increasing the ability and speed of the organization to deploy and operate AI/ML solutions at scale.

3. The ML Life Cycle

We capture the overall steps of the ML life cycle in Figure 3. Though, there are different variations of the AI/ML life cycle if we ask different vendors (see (Digging into AWS SageMaker - First Look, n.d.) (Davidson, n.d.) (Machine Learning Life Cycle, n.d.) for some examples), overall, we can break it down into the following five high-level components:

1. **Problem Definition including Goals and Metrics**
   This is usually performed by business or product owners.
   a. Outcome: a business objective that requires ML with a measurable metric and outcome to be optimized. In general, solutions in which we would need very complex heuristics and rules are often difficult to solve by traditional software and business logic, and the ML approach of searching for a specific set of parameters that optimize the desired objective is easier and more robust. Defining a target variable is critical. This can be something as

simple as a specific label on the dataset or a proxy metric. Lastly, the accuracy or measurable outcome is required because without it, the efficacy of the model would be unknown, and improvement would be impossible to calculate for the ML algorithms.

2. **Data Ingest & Storage**
   This step is usually the responsibility of software and data engineers.
   a. Ingest: Capture the data in the source system, ideally using well-defined and validated schemas while apply necessary governance tags.
   b. Streaming Processing: Route and process streaming data and make it available for real-time consumption while applying necessary hashing and de-identification processes.
   c. Storage: Store data in the data lake or a database.
   d. Governance: Apply the appropriate privacy and governance policies to both data in transit and data at rest.
   e. Metadata: Capture metadata in a metadata store to enable feature discovery in the next step.

3. **Data Analysis & Feature Engineering**
   This step is often performed by ML engineers, researchers, and analysts.
   a. Discover: Find data sources and existing code that can assist in the task
   b. Explore: Analyze and visualize data to identify potential useful features and patterns in data.
   c. Clean: Data cleaning can be a few to hundreds of steps depending on the state of the dataset (part of feature engineering).
   d. Prepare: Join and transform data to create appropriate features for model training (also part of feature engineering).
   e. Validate predictor (aka target variable): For supervised models this is critical for the outcome and accuracy measure. Care also needs to be taken here to ensure enough target variables represent the data to reduce bias. These labels may need to be generated programmatically, derived, or crowdsourced with something like a Mechanical Turk.

4. **Model Training & Evaluation**
   When most people think about ML, this is the step that usually comes to mind.
   a. Code: Develop model training code with rapid feedback.
   b. Collaborate: Review others' codes and validate model results, techniques, and assumptions.
   c. Track: Evaluate and track performance of experiments to select best models and make experiments reproducible.
   d. Train the model: Run the actual optimization code, which can include the need for distributed computing frameworks and may include hyperparameter optimization.

5. **Data Pipeline and Model Deployment and Operations**
   a. Deploy: Release the model artifact to a production or reliable environment.
   b. Serving: Expose the model as a service or embed the model in an event pipeline so predictions can be requested or made.
   c. Validation: Perform A/B tests and validate model performance in production.
   d. Monitor: Track and record metrics for model performance for retraining if necessary.
   e. Manage: Deploy retrained models without customer impact.

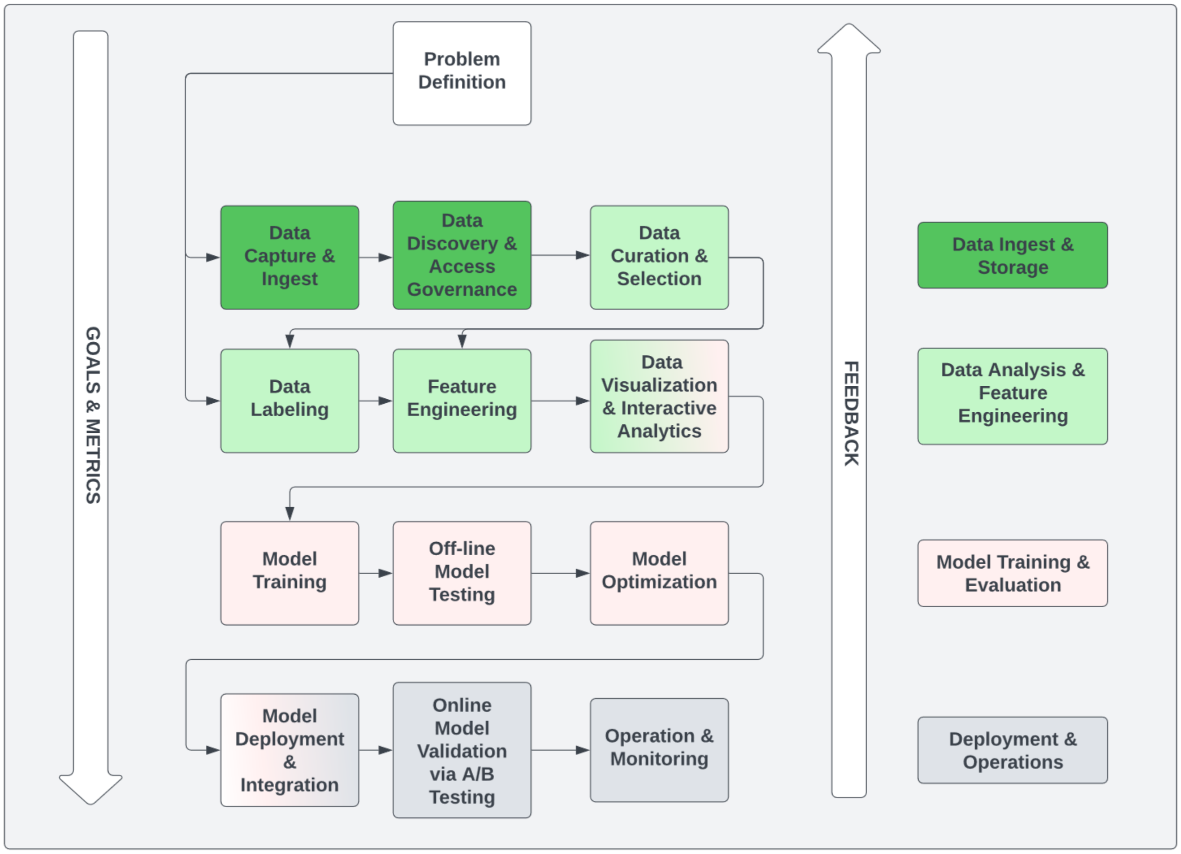


Figure 3 - The AI/ML Life Cycle

Comparing this to the software development life cycle (SDLC) as depicted in Figure 4, we can observe that the processes are similar and that at the high-level many activities in these two life cycles overlap or are simply the same. Understanding the similarities is critical to ensuring existing DevOps teams can transition to the AI/ML development aka software 2.0 (Karpathy, n.d.) model successfully. At the same time some of these items require specific consideration such as operations and processes related to features and models, and platforms supporting the operations and processes.

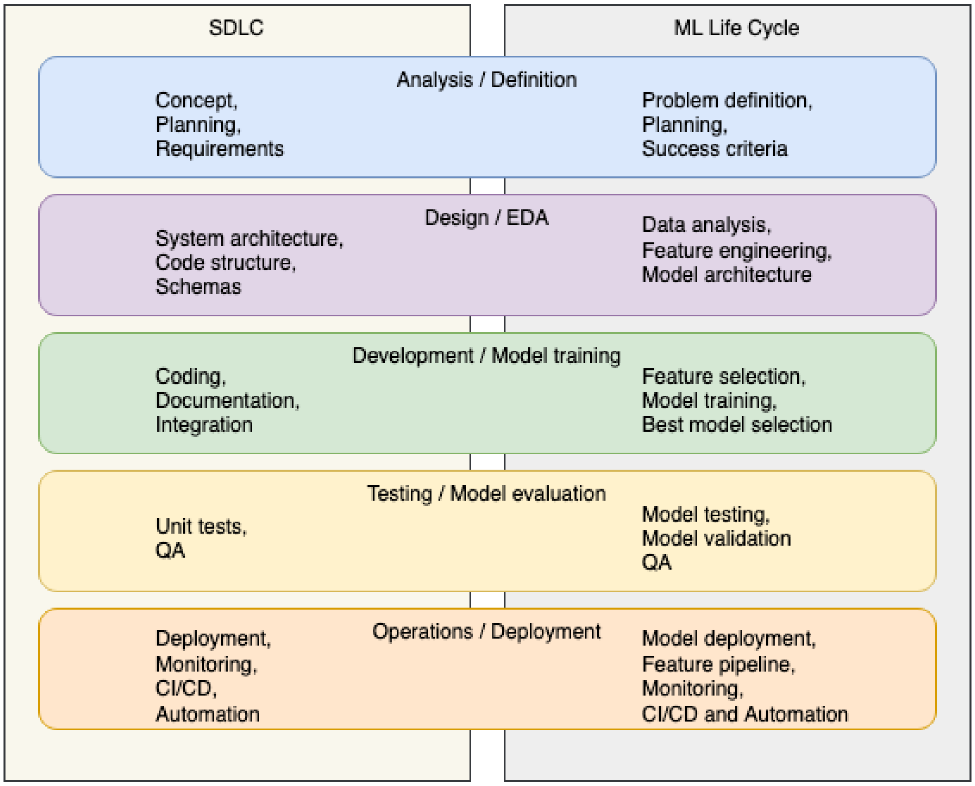


Figure 4 - SDLC and ML Life Cycle Commonalities

3.1. Features

Those familiar with software and ETL (extract, transfer, load) may consider features as simple or not as important as the model itself. This thinking is incorrect as the model is nothing without its features. In addition, features are the most powerful opportunity for domain experts to incorporate their knowledge into the machine learning model. One common challenge is that teams use different environments and platforms during the model training and inference stages, which can lead to problems if the feature engineering pipelines in production are not identical to the ones used during model training, testing, and validation. The feature store concept abstracts this difficulty and ideally this should simplify deploying the researcher's code to production in a way that can be easily supported by the MLOps team.

Feature engineering can be computationally intensive as well, which needs to be taken into consideration before production rollouts. Take for example an image of a kitten in grass shown in Figure 5.

Figure 5 - Kitten in Grass
(from https://unsplash.com/photos/RCfi7vgJjUY)

This image is 1567 x 1045 pixels. For each pixel value in the image, the value needs to be type converted, normalized, and encoded in the same exact manner that was performed during testing and training. Depending on the rate of images or requirements for the images (may need to be cropped, have bounding boxes, etc.) this can get computationally expensive. All these operations need to be considered when stress testing infrastructure, scaling, and monitoring production systems. Some teams may not be familiar with image, video, audio data or even sparse data structures. It is good to keep in mind that these differing types of datasets should be considered as well as the specific challenges for models.

QA for feature development is remarkably similar to traditional QA with the added step of exploratory data analysis (EDA). EDA allows the researcher to pick which features (fields, attributes, etc.) provide the most benefit for the model. This code is akin to ETL combined with a set of normalizations and transformations specific to the model being trained. For example, after desirable attributes have been chosen for a specific use-case, categorical features may be embedded into vector spaces via one-hot encoding or pre-trained embeddings to make them accessible to machine learning models.

3.2. Models

ML models may feel a bit more familiar to DevOps teams who are used to deploying code artifacts or compiled binaries. The ML model itself is an immutable artifact that must be versioned, tracked, stored, and deployed like other binary artifacts such as software releases. Model registries make this easier and are a close analog to a code artifact repository manager.

One key difference is that a software release is simply the software itself, while for ML models we also have to consider the data pipelines and that data input to the model is non-deterministic. As part of the MLOps platform, we need to monitor the inputs and outputs of these models in production to detect drift in the data distribution which gives an indication when to retrain and redeploy the ML models.

Traditionally, software is not being "taught" anything and its behavior is deterministically determined by the rules and heuristics captured in the code. This linear process is captured in the top-half of Figure 6, while the bottom half indicates how we split the data sets into train, validation (CV), and test data sets, and how the development and testing process is much more complex, iterative and non-linear due to the introduction of the data components.

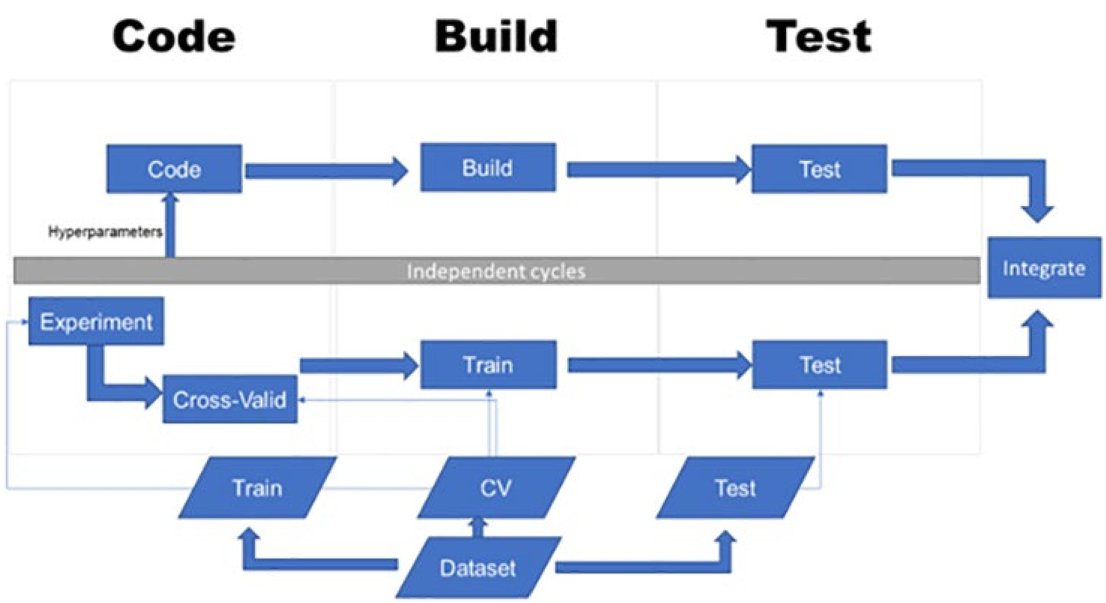


Figure 6 – DevOps vs MLOps Pipeline

3.2.1. Model Improvement

As described before, when training a model we are searching through the parameter space encoded by the model architecture to find the set of parameters that optimizes a specific metric. Improving ML models requires understanding the relationship between the changes in the training, validation and test data sets, and the changes to the model that we can make. For example, if the model is a classification model, do you want to improve the precision or recall, or perhaps some other metric? A great place to start is Andrew Ng's talk "Artificial Intelligence is the New Electricity" (Ng, n.d.) which is nicely summarized by Kevin Zakka (Zakka, n.d.). There is no single metric or gold standard that can be applied to every model apart from "train, test, analyze and repeat".

Continuous improvement of models is critical to the success of a project, team, or company. Given the need for continuous improvements, the development of AI/ML solutions can be broken down into two different phases: first, the functional integration of the solution into the larger application including setting up the data collection and processing pipelines ,which can usually be done in a relative short and well-defined period of time; followed by an iterative qualitative improvement of the model which can continue for the duration of the software life cycle.

3.3. Model Accuracy

3.3.1. Supervised

ML models can only be as accurate as the data used to train and validate them. Traditionally, this is done using labeled data to train and test against. These datasets are usually manually annotated, which is very time intensive but tends to lead to the most accurate results. Ideally this annotation or labeling step is assisted by the current machine learning models in an active learning framework such as SageMaker Ground Truth (Amazon SageMaker Ground Truth, n.d.), in which the algorithm indicates which examples should be annotated with labels to help the model the most.

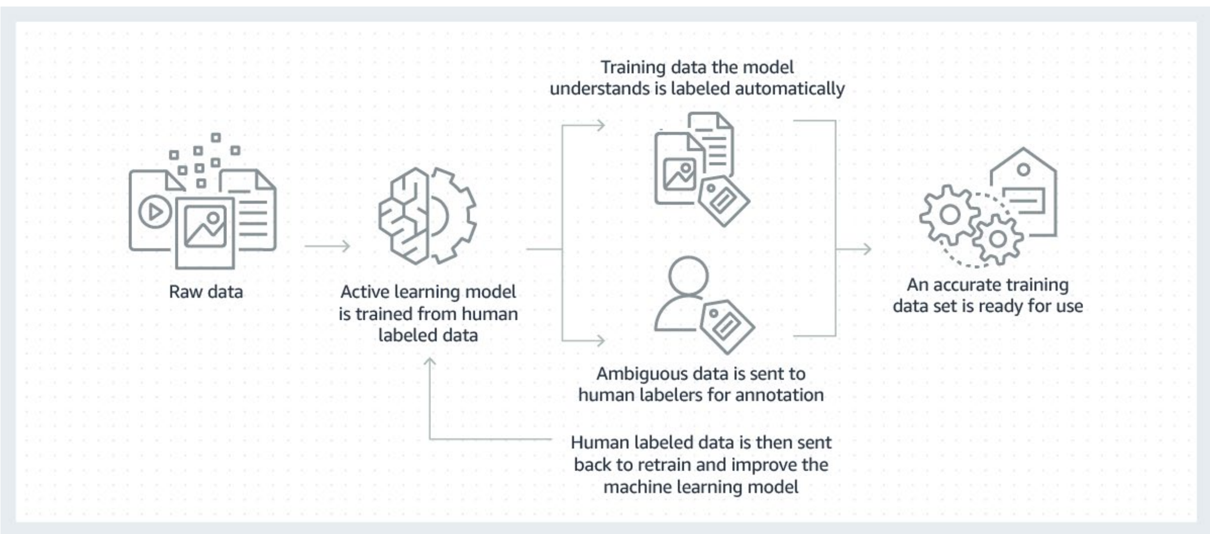


Figure 7 – AWS SageMaker Ground Truth Framework

Due to the limited number of records in some manually annotated datasets, augmentation of these data sets using tools such as Ground Truth can improve accuracy by prioritizing annotation of records for which the predictions are ambiguous or are of poor accuracy. These techniques are powerful for supervised training with labeled data, but how can the accuracy of unsupervised or weakly supervised models be measured?

3.3.2. Weakly Supervised

For weak supervision, or when model training is desired and not enough accurate training or labeled data exists, it may be possible to take advantage of implicit structure within the data sets. A tool to perform this technique (sometimes called "data programming") began as the Stanford Snorkel project and has evolved into a full open-source community and now a commercial product (Snorkel, n.d.).

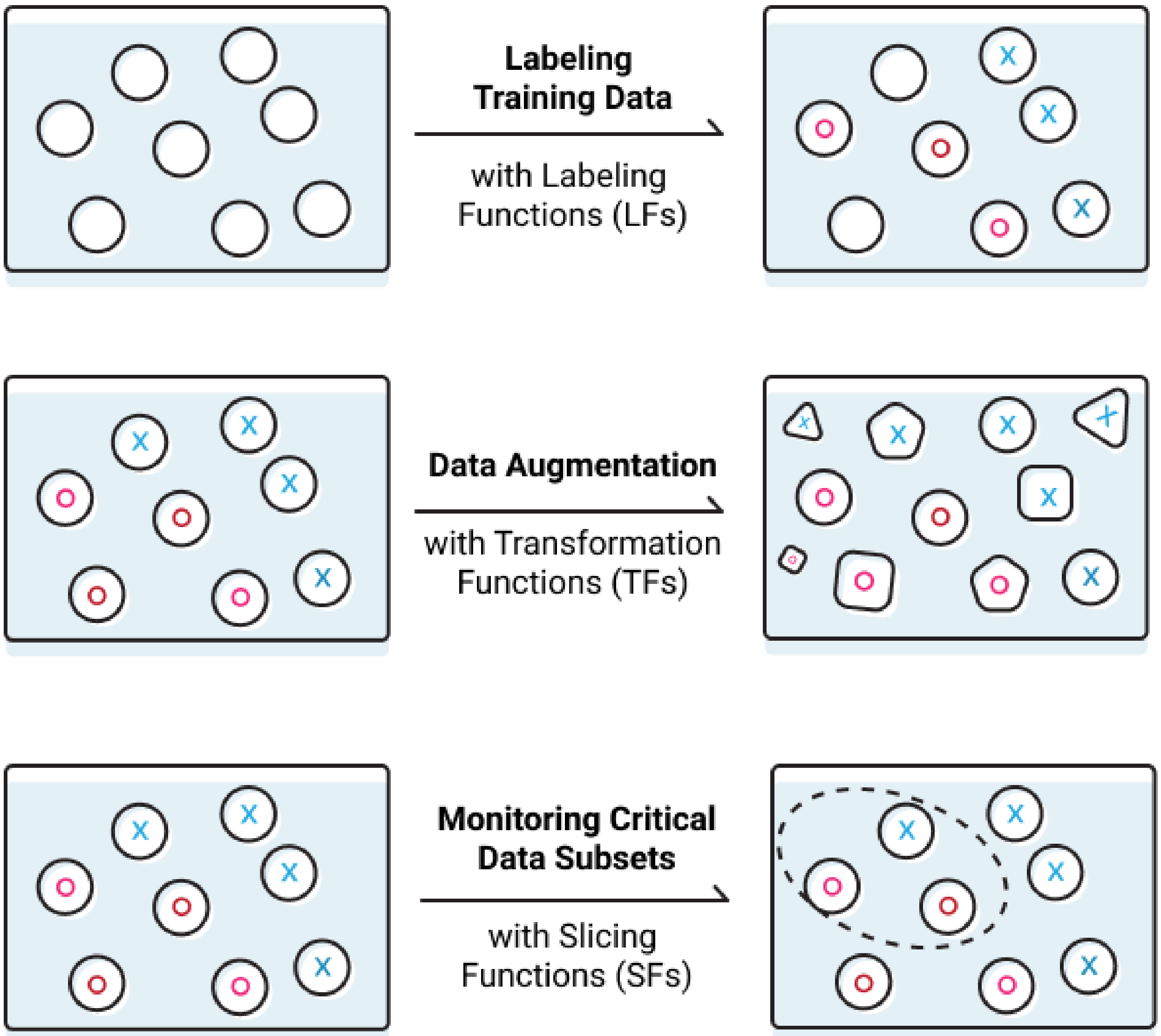


Figure 8 – Snorkel Overview

A system using weak supervision takes advantage of the fact that even data that is only partially labeled correctly can be useful as long as we know that we should trust that data less than more accurately labeled data. In addition, many of the modern deep learning network models (e.g., the large language or computer vision models) even operate well on unlabeled data because they are incorporating so much data into the training process that the algorithms are able to automatically extract the structure from the data even without labels.

What are some techniques to evaluate the quality of the data used as input into the models? The Stanford Dawn project (Paroma Varma, n.d.) describes utilizing provenance and data lineage to debug issues with the model. Here, they used Snorkel and other weak supervision systems to debug models based on features post-transformation. Other tools such as Google's What-If tool (Wexler, n.d.) can also help with the interpretation of the trained models and debugging their performance.

3.3.3. Model Bias

While stating computers are not biased is technically true; the data that is being used to "teach" the model about the world is often unfortunately not an accurate representation of the desired behaviors. There have been many highly visible examples of bias from the ProPublica "Machine Bias" (Julia Angwin, n.d.) criminal prediction article to the many studies on issues with facial recognition (Hao, n.d.) which can have far reaching effects. Some simple yet powerful techniques to reduce bias are to first look at the data and try to determine if enough data exists for each differentiator. Choosing the most appropriate model for the problem and even reducing accuracy in favor of balanced results can help. In MLOps, continuous monitoring of these models in production is critical to analyze how the model is performing and if changes are required. Once the model is deemed ready to begin its usefulness it must be released.

4. ML Platform Requirements

Armed with some background about MLOps, life cycle, and model improvement it is time to define what ML engineers and researchers require from an ML platform. Note that many products currently exist to tackle part or many of these various requirements, but to date no one platform encompasses them all.

4.1. General Consideration

The platform should;

- Utilize CPU and GPU resources in a transparent and cost-efficient manner both in the cloud and on premise;
- Integrate with existing authorization and authentication mechanisms specific to the company or organization; and
- Version and track code and model artifacts that are versioned and tracked (preferrable in a familiar source code repository and model store).

4.2. Requirements for different Personas

Different users and stakeholders of the system will require separate outcomes from the platform.

4.2.1. Business or Product Owner

From the business or product owner perspective, the platform should provide metrics to measure success of projects in an easy manner.

4.2.1. Data Scientist or Researcher

From the perspective of the data scientist or researcher, several requirements exist that are hopefully satisfied by the platform. At the very least, the requirements should be considered and satisfied by separate tools and products to be later integrated into a more seamless experience.

Optimally, data scientist or researcher should have:

- IDEs and remote execution environments (i.e., Jupyter Notebook) to develop and document the code to control the feature generation and models training.
- Processes (i.e., Anaconda, virtualenv, etc.) to simplify library installation or utilize bundles that contain the necessary libraries (such as pandas, NumPy, scikit-learn, TensorFlow, etc.).
- Tools and products to aid in feature generation and automatic training (i.e., Databricks AutoML, H2O, DataRobot, SparkBeyond, etc.).
- Access to tools that aid with model interpretation.
- The ability to view training results quickly to shorten the feedback cycle.
- A way to train models at scale utilizing both on-prem and cloud resources.
- A storage platform that minimizes storage and data transfer costs and time for training, testing, and exploratory data analysis.
- A compute platform that utilizes distributed computing and auto-scaling to minimize cost for training and scoring.
- A staging or production-like environment to deploy and validate feature engineering flows and models.
- Processes to track artifacts, code, and datasets as well as results and metrics for easy re-training and experiment replication.
- A UI or visualizations to quickly analyze performance of different models.
- Methods to identify specific examples of misclassifications or anomalous results outside the training distributions.
- Tools that aid in the sharing, collaboration, and review of code, models, and datasets.
- Standardized processes and environments to ensure portability of models and processes.

4.2.1. Data or ML Engineer

It is not enough however to focus only on the data scientists and researchers within the team or organization. The ML and operations engineers have specific requirements as well. These should include:

- Environment standardization to simplify ease of deployment and promote consistency.
- Tools to rapidly and easily deploy or update models in production.
- Methods to determine and automatically deploy best performing models.
- The ability to deploy models utilizing multiple frameworks, languages, and technologies (i.e., Docker-based, serverless, streaming operator or UDF, many more.
- To minimize the latency of inferences and predictions at scale.
- To optimize throughput to models via batching and GPU offload when possible.
- The ability to utilize production traffic for system level and performance testing.
- Alerts when latencies and prediction statistics deviate from normal patterns.
- Alerts or automatic retraining (if necessary) when prediction or inferences consistently decrease with regards to the appropriate accuracy metric.

- Logs and distributed tracing of the feature engineering data transformations and model scoring.
- Lineage of the features as they flow through the system to the model.

With the requirements gathered from product, research and engineering perspectives, the needs of the platform itself must be addressed. From the data or ML engineering perspective, the platform should:

- Allow for easy use and discovery of static (files) and dynamic (streaming) data sets.
- Encourage re-use of past research and models for exploration or novel solutions via search, documentation, and model stores.
- Allow for recipes, templates and techniques for data cleaning and feature engineering to be shared and accessed easily.
- Provide a central (or seemingly central) data and feature store to promote dataset and feature reuse with an added benefit of minimizing the duplicate feature engineering computations.
- Allow for temporal constraints to be placed on features for update, recompute or expiration to reduce feature rot.
- Provide mechanisms to track changes in data and feature to warn against ingest and feature engineering issues.

5. Enabling MLOps with ML Platforms

The focus of MLOps on automation and standardization makes it necessary to develop a framework that supports the development and operations effort by automating any repetitive steps.

Platforms can make the deployment of ML solutions more efficient by providing generic, reusable components that simplify the ML workflow and make it easy to do the right thing every time. A standardized platform also encourages cross-team and cross-company sharing of feature data and models in addition to just sharing data as done by pure data platforms.

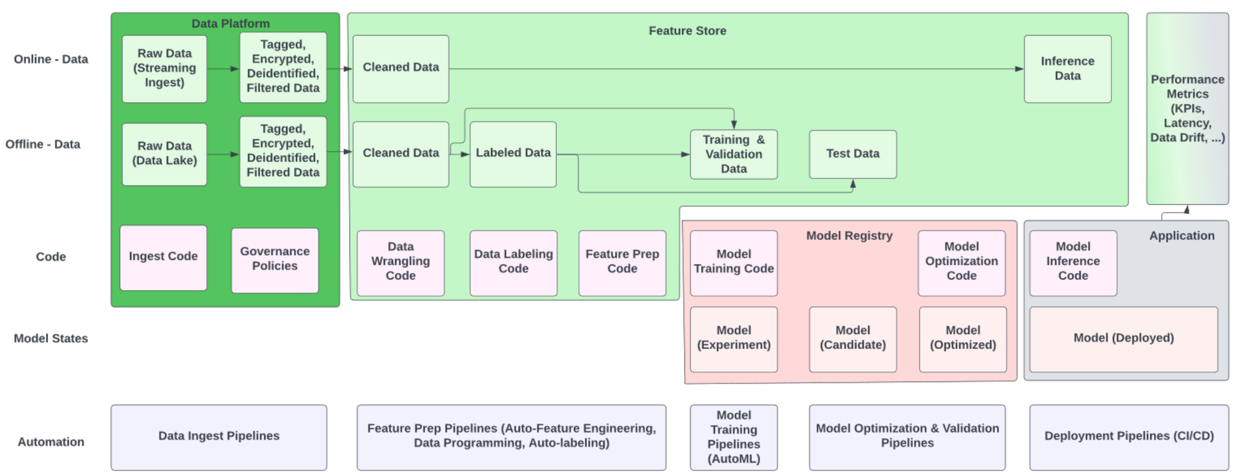


Figure 9 - AI/ML Platform Components

5.1. Common Functions of ML Platforms

While the AI or ML Platform may become more ubiquitous, it is helpful to understand the different components of the overall platform and the benefits a platform can provide, such as captured in Figure 9:

- **Data Platform for Ingest, Storage, Discovery & Governance**
  Often one of the main problems for MLOps is the easy ingest of and access to the necessary data for the AI/ML models while keeping costs down and adhering to the applicable contractual and regularity privacy and governance requirements.

- **Feature Store Platform**
  Often features need to be transformed and normalized before they can be passed into the model. ML platforms are often providing subsystems to stream features into models such as messages buses with serverless functions. Examples of this and similar functionality can be found in systems such as AWS Lambda, Apache Spark, and Kubeflow Pipelines.

  When stateful features or disparate features which need to be joined are required by a model it is common to see them stored in a feature store that is abstracting the large off-line data sets often stored in S3 and used for model training and the on-line data used for real-time model inference often accessed via a high-performance cache or database layer.

  Example systems not specifically designed but often used for this purpose are RDBMS or NoSQL databases, caching DB layers such as Redis, and the feature stores offered by Feast, Tecton, AWS, Google Cloud Platform and Databricks.

- **Model Registry and Model Server**
  To help with the reproducibility of AI/ML models we want to capture all of the relevant code and training information together with the model in one place. This place is called the model registry and it acts as an abstraction boundary between the model training and the model inference processes. Example model registries are MLFlow and the AWS model registry.

  To serve a model means to expose it via a common interface such as HTTP or gRPC as a service that can be utilized by other systems, users, and processes. Examples of model serving are AWS SageMaker, KFServing, Seldon Core, BentoML, Azure Machine Learning, Nvidia Triton and many more.

- **Development Platform** (not depicted, used to develop the code artifacts in the diagram)
  Many platforms provide Jupyter-type or other environments to develop code in conjunction with the ecosystem of components provided by the platform. Examples of this are the Jupyter-hub built into Kubeflow and SageMaker Studio by AWS. One can also use standard software IDEs such as VSCode or PyCharm who are already popular with software developers.

6. Example ML Platforms and Systems

In 2016 the number of ML platforms and open-source solutions numbered in the dozens. Now there are hundreds of ML platforms and supporting projects encompassing open source and commercial offerings from large companies and startups alike. To get a sense of how large the landscape was even in 2018, Redpoint (Myers, n.d.) provided a wonderful visualization with roughly 280 tools and platforms. The Linux AI & Data Foundation currently contains already 44 open-source software projects in the AI and Data space and even CB Insights Top 100 AI Startups list is just showing the tip of the iceberg of solutions that try to address parts or all of the AI/ML life cycle.

While it is not feasible to cover every platform and technology in detail, we want to mention here several of the mature offerings that cover most of the AI/ML life cycle components and can be recommended as starting points for building an MLOps platform.

6.1. AWS SageMaker

Amazon's SageMaker ecosystem and its set of components are comprehensive. They provide collaboration via SageMaker Studio, interactive feature preparation and generation along with model training and deployment. Note that the SageMaker, Lambda and other components are fully integration with the rest of Amazon Web Services .

6.2. TensorFlow TFX

TensorFlow TFX is Google's alternative to Amazon SageMaker. TFX offers large scale production environments for model serving, ML workflows, collaboration and of support for TensorFlow and Google Cloud's Tensor Processing Units (TPUs).

6.3. Azure Machine Learning

While some of Microsoft's ML Platforms are a bit newer to the scene, they offer end-to-end MLOps and many user-friendly tools for ML. AzureML supports MLflow, Kubeflow, and more as well as VSCode and Jupyter for collaboration.

6.4. Databricks

Databricks was started by the inventors of the Apache Spark programming framework and they added an implementation with Jupyter notebooks as the main interface. They have since grown to cover many more aspects of the ML space with the model registry MLFlow, a feature store on top of the Delta storage format, comprehensive data and model governance via Unity Catalog, and SQL analytics and serverless inference capabilities. The core offering is not free or open source, but some components are (such as MLFlow, Spark, Delta), and Databricks is available across all 3 of the major cloud platform providers.

6.5. H2O

Initially H2O was known for their high-performance implementation of machine algorithms, but now their tools offer automatic feature engineering, AutoML, a feature store, model serving and other ML platform capabilities. Some of their offerings are open source but the majority are not.

6.6. Kubeflow

Kubeflow is a Kubernetes based AI/ML platform that offers open-source versions of most of the major ML life cycle components. Kubeflow has been around for a few years now and offers a comprehensive feature set most users and operations folks may be interested in leveraging.

7. Conclusions

There are many ML workflow and platform systems with new ones continuously being added. Choosing among them can be difficult, but no matter which platform, technology, or workflow is chosen, good practices must be followed. In this work we described both the similarities and differences between the software development and machine learning life cycles and provided guidance for what factors should be considered when building AI/ML platforms to enable MLOps at scale. The described best practices will greatly increase the success and outcome of the ML workflow system, which we hope will accelerate the adoption of AI/ML solutions in the industry.

8. Abbreviations and Definitions

8.1. Abbreviations

AWS	Amazon Web Services
CI/CD	continuous integration / continuous deployment
CPU	central processing unit
DNN	deep neural network
EDA	exploratory data analysis
ETL	extract, transform, load
HTTP	Hypertext Transfer Protocol
GPU	graphics processing unit
gRPC	gRPC remote procedure calls
IDE	integrated development environment
IT	information technology
ML	machine learning
MLQA	machine learning quality assurance
QA	quality assurance
RDBMS	relational database management system
SDLC	software development life cycle
SQA	software quality assurance
UDF	user defined function in Apache Spark

8.2. Definitions

AutoML	automated machine learning – to automate various repetitive or time-consuming tasks related to feature selection, model selection and model training
DevOps	combination of software development and IT operations
MLOps	machine learning operations
model	generally referred to as a machine learning model, artifact, conditionals or set of constants or coefficients generated from input data or other patterns

References

AI 100: The most promising artificial intelligence startups of 2022. (n.d.). Retrieved from https://www.cbinsights.com/research/report/artificial-intelligence-top-startups-2022/

Amazon SageMaker Ground Truth. (n.d.). Retrieved from https://aws.amazon.com/sagemaker/groundtruth/

Davidson, A. (n.d.). *MLflow: Infrastructure for a Complete Machine Learning Life Cycle*. Retrieved from https://www.slideshare.net/databricks/mlflow-infrastructure-for-a-complete-machine-learning-life-cycle

DevOps, Wikipedia. (n.d.). Retrieved from https://en.wikipedia.org/wiki/DevOps

Digging into AWS SageMaker - First Look. (n.d.). Retrieved from https://engineering.upside.com/digging-into-aws-sagemaker-first-look-90975d80cd87

Edge Computing at Chick-fil-A. (n.d.). Retrieved from https://medium.com/@cfatechblog/edge-computing-at-chick-fil-a-7d67242675e2

Evans, B. (n.d.). *Ways to think about machine learning*. Retrieved from https://www.ben-evans.com/benedictevans/2018/06/22/ways-to-think-about-machine-learning-8nefy

Hao, K. (n.d.). *Making face recognition less biased doesn't make it less scary*. Retrieved from https://www.technologyreview.com/2019/01/29/137676/making-face-recognition-less-biased-doesnt-make-it-less-scary/

Jez Humble, G. K. (2018). *Accelerate.* IT Revolution Press.

Julia Angwin, J. L. (n.d.). *Machine Bias*. (ProPublica) Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Karpathy, A. (n.d.). *Software 2.0*. Retrieved from https://karpathy.medium.com/software-2-0-a64152b37c35,

Linux Foundation AI & Data Projects. (n.d.). Retrieved from https://lfaidata.foundation/projects/

Machine Learning Life Cycle. (n.d.). Retrieved from https://www.datarobot.com/wiki/machine-learning-life-cycle/

Myers, A. (n.d.). *Memory Leak - Medium*. Retrieved from https://medium.com/memory-leak/introducing-redpoints-ml-workflow-landscape-312ca3c91b2f

Ng, A. (n.d.). *Artifical Intelligence is the New Electricity*. Retrieved from https://www.youtube.com/watch?v=21EiKfQYZXc

Paroma Varma, B. H. (n.d.). *Using Provenance to Debug Training Data for Software 2.0*. Retrieved from https://dawn.cs.stanford.edu/2018/06/21/debugging/

Snorkel. (n.d.). Retrieved from https://www.snorkel.org/

Wexler, J. (n.d.). *The What-If Tool: Code-Free Probing of Machine Learning Models*. Retrieved from https://ai.googleblog.com/2018/09/the-what-if-tool-code-free-probing-of.htm

Zakka, K. (n.d.). *Nuts and Bolts of Applying Deep Learning* . Retrieved from https://kevinzakka.github.io/2016/09/26/applying-deep-learning/

SCTE STANDARDS JOURNAL

An Emerging Alternative for Meeting Zero-Emissions Goals

Fuel Cell EVs and Hydrogen Fueling

Letter to the Editor prepared for SCTE by

Seth S Terry, CEO, New Day Hydrogen
4285 S Bellaire Circle
Englewood, CO 80113
seth@newdayhydrogen.com
303-946-8710

Table of Contents

Title **Page Number**

Table of Contents 94
1. Introduction 95
2. Discussion 95
   2.1. Emissions? Why Care? 95
   2.2. Transitioning to Zero Emissions 95
   2.3. ZEV Support Infrastructure 96
   2.4. ZEV Complements ZEV 96
   2.5. A Viable Path to Building Hydrogen “Cheggs” 97
3. Conclusion 97
4. Abbreviations 97

1. Introduction

Increasingly, major corporations including those in cable and telecommunications face pressure to develop zero-emissions goals. Today, this pressure derives primarily from investor engagement by way of institutional leaders such as BlackRock, State Street, and Vanguard. However, the specter of regulation also looms large from agencies like the US Securities Exchange Commission *via* its proposed climate disclosure rule. As a consequence, discussions over reducing greenhouse gas (GHG) emissions have long since entered the board room. And with transportation now accounting for the largest portion of GHG emissions in the US, corporate fleets represent an obvious place to look for cuts.

For many, the obvious answer for these emission reductions is to start the transition to battery electric vehicles (BEVs). However, though doing so may prove relatively easy for light-duty vehicles with devoted overnight space for charging, fleets of heavier-duty vehicles will struggle operationally with the dwell time required to replenish range. For this reason, fleet owners may do well to consider an alternative to BEVs that can simultaneously meet zero-emissions goals ***while*** providing the convenience of fueling: fuel cell electric vehicles (FCEVs). And, although this type of electric vehicle (EV) is only beginning to enter the market, many expect FCEVs soon to become the dominant choice of heavier-duty fleets.

2. Discussion

2.1. Emissions? Why Care?

Even beyond investor concerns and regulatory threat, reducing emissions helps avoid negative health and climate effects to all of us, whether in urban or rural areas. For city dwellers, the US Department of Transportation recognizes that living near highways and major thoroughfares increases our risk of respiratory effects from ground-level ozone, oxides of nitrogen, and fine dust. Such effects include asthma, pneumonia, bronchitis, and heart conditions. And in this regard, we should note that heavier-duty vehicles represent a special concern since diesel particulates pose a heightened risk for lung cancer.

But those living outside of cities are not spared either, since fossil fuels emit GHGs, which many believe are a root cause of extreme weather events such as drought, wildfires, and flooding. As many of us know well, these incidents can visit horrible economic consequences on both infrastructure and people within our own communities. In 1800, the beginning of the Industrial Revolution, carbon dioxide (CO2) concentrations were 283 parts per million (ppm). When some of us mid-lifers entered the scene 167 years later, CO2 concentrations had risen 37 ppm to 320 ppm. In 1998 (when some of us started families of our own), CO2 was already 46 ppm higher at 366 ppm. And now, only 24 years later, we have reached 421 ppm – a full 55 ppm higher yet! Clearly, GHG emissions are accelerating and posing a generational risk.

2.2. Transitioning to Zero Emissions

As stated, transportation comprises the largest part of today's energy emissions in the US, now eclipsing even electricity generation. And as a result, fleets present an obvious target for businesses to reduce GHGs through transition towards so-called "zero emissions vehicles" or ZEVs for short. As indicated, ZEVs exist in two flavors: BEVs and FCEVs, both of which are electric.

But whereas BEVs store and deplete electrons in battery banks, FCEVs store and deplete gas – specifically hydrogen – in tanks. And whereas BEVs replenish range while idle and attached to a charger, FCEVs fuel up with a short visit to a station.

Each ZEV type can serve different but complementary functions in much the same way that gasoline and diesel energize different but complementary vehicles today. Speaking generally, light-duty applications skew towards gasoline whereas heavier-duty applications favor diesel. As we transition to ZEVs, light-duty applications will tend towards batteries while heavier-duty applications will tend towards hydrogen. The latter enables vehicle range to be replenished in minutes rather than requiring hours of chargingtime.

2.3. ZEV Support Infrastructure

As indicated, FCEVs require hydrogen fuel. However, today there is almost no hydrogen for vehicles outside of California. And, although electricity is widely available, BEVs too face their own challenges since charging large numbers of these vehicles requires significant grid upgrades. Therefore, just as we constructed stations for gasoline and diesel (conservatively estimated at a cost of $70B for approximately 145,000 stations in the US), so too will we require new infrastructure to support ZEVs.

In this regard, BEVs clearly hold the early advantage given the universal presence of electricity. Just about every business in the US has adequate power to meet daily requirements – and likely can tolerate the marginal extra demand required for a small number of BEVs. However, charging numerous BEVs simultaneously or trying to do so faster (for example with Level-III charging) may prove difficult. The electric distribution network may be inadequate to supply the power needed for safe charging within the desired timeframe. This substantial infrastructure cost is the major challenge for widespread deployment of BEV charging. Potential additional costs include trenching, coring, and rewiring needed across parking lots, inside structures, and under streets in dense urban settings, or between small rural towns with limited electrical supply.

In contrast, FCEVs and hydrogen claim no early advantage. Outside California, almost no infrastructure exists for hydrogen fueling – making it impossible to replenish even a small number of FCEVs using existing infrastructure. For this reason, people often refer to the “chicken and egg” challenge around initiating hydrogen infrastructure. In other words, it is difficult to invest in infrastructure without demand from vehicles, but vehicles won’t be available without fueling infrastructure to support them. Put another way, as quipped a friend working at the National Renewable Energy Lab, we need to build “cheggs” – and better yet cheggs of modest size to accommodate fueling at early stages of fleet transition.

2.4. ZEV Complements ZEV

Both ZEV types suffer from their own infrastructure weaknesses. However, both types of ZEVs also boast respective strengths. These strengths and weaknesses complement one another very well regarding strategies for building out infrastructure to support transportation’s transition. Specifically, batteries have worked well to seed early adoption for EVs in general. And fittingly, the lion’s share of early adopters have included light-duty, passenger vehicles – most of which park and charge overnight in devoted garage spaces and are used most effectively for shorter trips.

But batteries struggle to fit ZEV applications intolerant to extended down-time for charging or where space or electrical resources are limited for serving large numbers of vehicles. Similarly, BEVs struggle when range exceeds battery capacity, when cold or hot weather decrease that battery range, or where the weight of batteries cuts into cargo capacity. For these reasons, many sector watchers accept that heavier-duty vehicles and fleets, often today's users of diesel, will generally move to hydrogen fueling when such becomes available.

2.5. A Viable Path to Building Hydrogen "Cheggs"

Since existing infrastructure for hydrogen fueling remains largely absent, it is necessary to address just how FCEV adoption could be accomplished. We need to keep three considerations in mind. First, initial station infrastructure must accommodate early-adopters starting with small numbers of FCEVs as they begin their fleet transition. Second, at the outset of FCEV adoption, most fleets cannot rely upon a fueling network to support their operations. And third, despite desires to reduce GHG emissions, fleet owners will not tolerate radically expensive fuel.

Hydrogen production from electricity (i.e., electrolysis) offers distinct advantages over other methods of production. This method "cracks" water molecules without emitting carbon dioxide or other GHGs at the point of production and, if the required electricity is derived from renewable sources, produces 100% green hydrogen. Further, electrolysis readily accommodates the three considerations mentioned above. There are multiple other advantages as well: First, electrolysis can occur across a broad range of production volumes, including those at smaller, early-adopter scales. Second, electrolysis can occur easily and with modest footprint (e.g., 2.5 shipping containers) on-site for centralized fleets – and thus insulates early adopters from fueling from an outside network or relying upon a supply chain for delivery. Third, with appropriate pairing of hydrogen production and use, fuel pricing can be attained at competitive rates that show a glide-path over time towards parity with diesel. And finally, because electrolysis produces and stores hydrogen at a constant rate throughout the day (in contrast to BEV charging), it avoids peak demand charges for electricity during periods of rapid refueling.

3. Conclusion

Because electrolysis-based hydrogen fueling provides such versatility, it enables deployment of FCEV fueling immediately – even in advance of billion-dollar regional hydrogen hubs. The time for seeding hydrogen fueling has arrived and fleets need not await full fueling networks in order to begin their path towards meeting zero-emissions goals. The technology and approach are here for the taking. Now all that is required is the desire to proceed.

4. Abbreviations

BEV	battery electric vehicle
CO2	carbon dioxide
EV	electric vehicle
FCEV	fuel cell electric vehicle
GHG	greenhouse gas
ppm	parts per million
ZEV	zero-emissions vehicle

SCTE
a subsidiary of CableLabs®

140 Philips Road, Exton, PA 19341-1318 | 800.542.5040 | **scte.org**