Enhancing ISP Network and Service Optimization Through Causal Inference and Knowledge Base Development

A technical paper prepared for presentation at SCTE TechExpo24

Sebnem Ozer, Ph.D.
Distinguished Engineer
Charter Communications
sebnem.ozer@charter.com

Deependra Rawat
Principal Software Engineer
Charter Communications
deependra.rawat@charter.com

Phil Anderson, Charter Communications

Lei Zhou, Ph.D., Charter Communications

Jordan Waldroop, Charter Communications

Yablai Bougouyou, Charter Communications

Daniel Lynch, Charter Communications

Table of Contents

List of Figures

List of Tables

1. Introduction

The surge in applying Artificial Intelligence (AI) for network and service quality and efficiency optimization is undeniable. However, current AI techniques struggle to define cause-effect relationships. This limitation poses a risk when applying these techniques to a vast array of telemetry data without a solid knowledge base. While many Internet Service Providers (ISPs) have been integrating latency measurements into their operations tools, the analysis of latency and other QoS metrics is still a developing research area. Misleading ISPs to false or missed cause-effect relationships can lead to ineffective optimization methods. Therefore, while machine learning techniques are extensively explored to manage efficient and high-quality platforms, a potentially important aspect lies in establishing robust telemetry and knowledge-based systems.

In this paper, we analyze the latency test cases for Internet Engineering Task Force Low Latency, Low Loss, and Scalable Throughput (IETF L4S) applications over a Low Latency Data Over Cable Service Interface Specifications (DOCSIS®) network path and an internet network segment. We employ a two-step approach: firstly, analyzing latency in known bottleneck or unstable links, followed by estimating unknown network segments using end-to-end measurements. We then discuss major causes within these links by using different latency models. Through L4S streaming experiments and latency test data analysis, we discuss the limitations in conventional predictive AI and explore causal reasoning's efficacy. By testing various latency-inducing scenarios across network segments and using large-scale test data, we demonstrate the role of correct data collection and error identification.

Predictive AI excels in identifying correlations but cannot detect causal relations. Conversely, causal AI today demands a substantial knowledge base for effective analysis and is a less mature domain with limitations. This study discusses the causal inference for latency issues in ISP networks, proposing a simplification approach to identify major causal factors despite multi-segment network complexity. We believe a solid knowledge base on network access technologies such as DOCSIS, transport protocols such as L4S, and measurement methods will help early causal inference models in latency optimization systems. We then show how a reinforcement learning-based model can iteratively learn from experiment results to refine actions in resolving bottleneck issues with self-correction for systematic errors.

2. Latency Analysis of Tandem Access and Internet Networks for L4S Traffic

Different sources of latency in access technologies have been an active research area in recent years. Many ISPs started to include latency and jitter measurements into their operations tools for different use cases from troubleshooting to network optimizations. Idle latency and latency under load are common measurements used by ISPs [1]. With the introduction of Low Latency DOCSIS, two sets of measurements-one for classic traffic and another for low latency traffic, such as IETF L4S traffic-can be measured and analyzed as shown in Figure 1. This type of measurement helps to discover large queues along the path, misconfigurations, and link and device issues. However, the end-to-end network is complex with many variables and network visibility cannot be scaled to every packet interaction in the whole footprint.

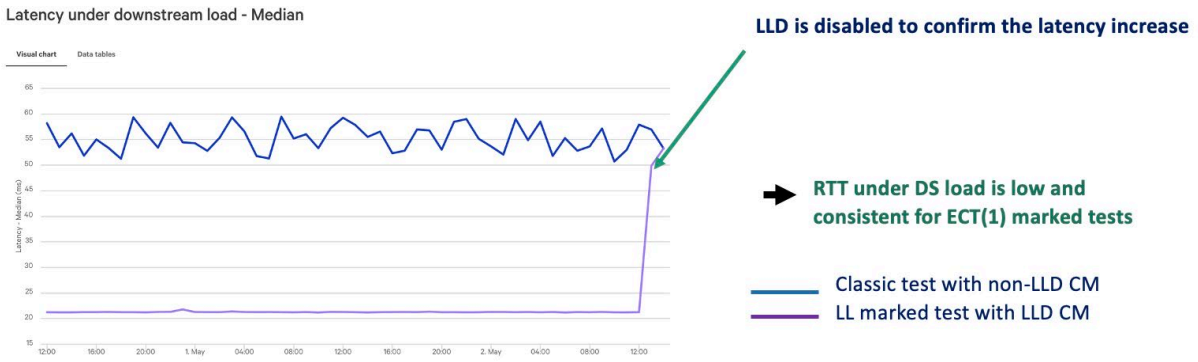


Figure 1 – Classic and Low Latency Measurement (SamKnows Data)

Tandem networks consist of multiple sequentially connected segments. In queueing theory, tandem networks are widely used to model end-to-end networks where packets are sequentially processed at each network segment with its own service and queue. A combination of direct measurement, statistical modeling, and analytical techniques is employed. Latency for each network segment is either actively measured using specialized tools or estimated using statistical or analytical approaches based on historical data and network characteristics. Each segment's latency is represented as a distribution to account for variability.

In simple cases, assuming independence between segments, the total end-to-end latency can be approximated by summing the mean latencies of each segment. Similarly, the total variance is calculated by summing the variances of each segment's latency distribution. Stochastic modeling techniques are used to model the probabilistic nature of packet transmission and delays. Additionally, new models have been proposed for networks where queue and service mechanisms are not independent between segments and where closed-loop feedback systems are required to model Transmission Control Protocol (TCP) and QUIC type flows adequately. Assumptions in terms of arrival and service distributions and simplification of the congestion avoidance algorithms are used to enable mathematical models.

The concept of tandem networks with simplified queueing functionalities can serve as the first step for a hierarchical analysis. The coarse data can be used to detect more visible issues reflected on the common measurements, followed by a higher resolution analysis of a more focused set. We created different test cases to test a multitude of speed tier rates, traffic conditions and DOCSIS parameters to confirm the concept of estimating latency contributions from different network segments. For this purpose, we used end-to-end latency measurements as well as DOCSIS congestion and latency metrics. The system can be used for estimating average, standard deviation, and median in networking terms—metrics like latency and quality of service (QoS) are relatively straightforward due to their frequency and regular occurrence. However, predicting rare and short-duration events such as microbursts poses significant challenges. These events, characterized by sudden spikes in network traffic, can disrupt services such as immersive and interactive applications, despite their infrequency, making them difficult to anticipate and mitigate.

In a tandem network scenario, where multiple segments affect overall performance, a generalized equation incorporating statistical measures of latency and QoS (such as mean, variance, etc.) can be expressed as:

Overall Performance=f(statistics of latency and QoS at each segment) Θ Probability of rare events

Here, $f(\cdot)$ represents a function that combines the statistical metrics (like mean latency, variance of QoS, etc.) across each network segment. The term "Probability of rare events (microbursts)" serves as a correction factor, reflecting the likelihood and impact of unexpected spikes in network activity. This equation encapsulates the challenge of managing both regular network performance metrics and the unpredictable nature of rare events in tandem networks in our analysis. We used this model to create network impairments that include both large statistics and rare events. With detailed L4S traffic generation tools as shown in Figure 2, additional metrics collected at the transport and application levels (including Explicit Congestion Notification (ECN) Capable Transport (ECT) metrics) enable mapping the QoS metrics among different network layers.

As shown below for a Low Latency DOCSIS case, in the first set of graphs, L4S traffic is the only traffic in the downstream direction for the subscriber and there is no bottleneck link in the end-to-end network. Figure 3 displays end-to-end results measured at application layer (with 100ms measurement intervals) while Figure 4 displays end-to-end results at network level (at each captured packet). DOCSIS metrics are also collected from the CMTS and CM devices. In these use cases, we assume that we don't have any measurement from the network segment before the DOCSIS network and that it doesn't support IETF L4S. In the actual deployments, passive measurement tools may provide results for certain traffic types and at certain resolution, but we are using the packet level measurements to analyze the system to gain knowledge base.

From the figures, we can confirm that the L4S traffic uses the bandwidth efficiently for the downstream link configured with 390 Mbps maximum sustained rate, while the 99.9 percentile one-way latency is less than 3 ms. The only Congestion Experienced (CE) markings happen when traffic comes more bursty based on the LLD configuration parameters, followed by adjustments of burstiness at the application. Four L4S flow statistics are presented to show the fairness among the flows in the ideal case.

Figure 2 – L4S traffic models : Top: Apple L4S QUIC; Bottom Left: Excentis ByteBlower L4S; Bottom Right: Linux TCP Prague

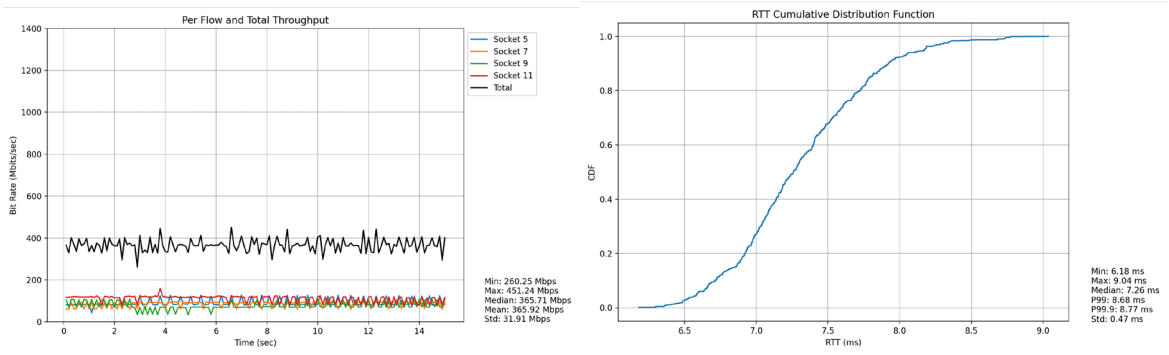
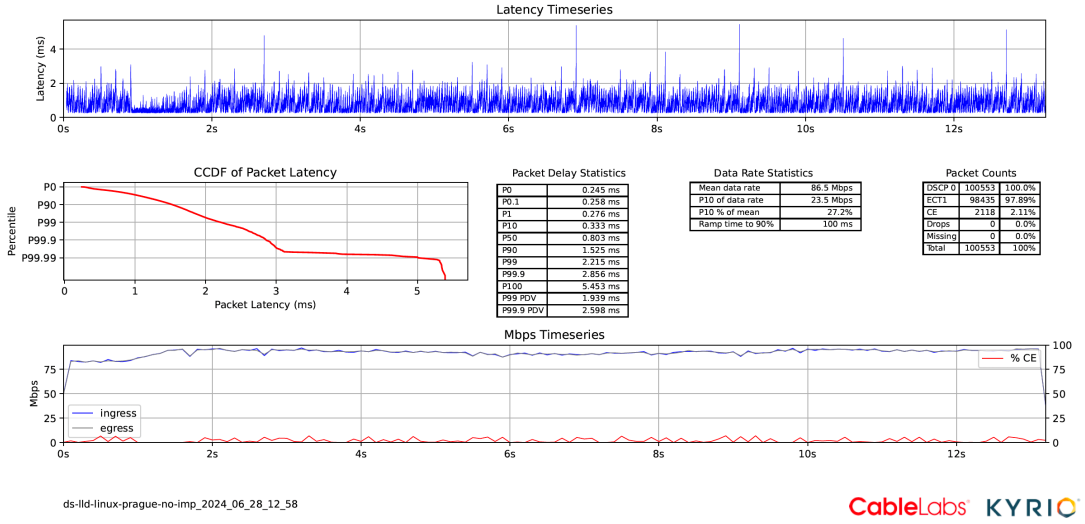
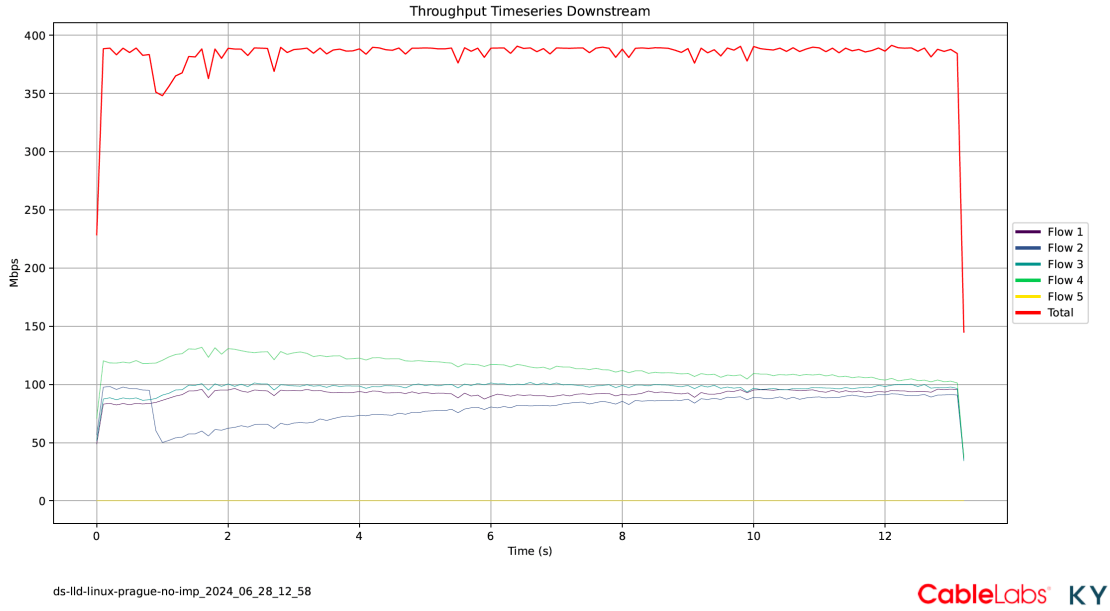


Figure 3 – L4S Traffic Results at Application Layer (100ms measurement interval) with no bottleneck network segment (L4S application benchmarking)



Aggregate_thruput_downstream.pdf



summary_table_downstream

Flow Number	Packet Delay P0 (ms)	Packet Delay P90 (ms)	Packet Delay P99 (ms)	Packet Delay P99.9 (ms)	PDV P99 (ms)	PDV P99.9 (ms)	Mean data rate (Mbps)	P10 of data rate (Mbps)	P10 % of mean	Ramp time to 90% Tput (ms)	Num Packets	Num NotECT	Num ECT0	Num ECT1	Num CE	Dropped Packets	Missing Packets
1	0.245	1.525	2.215	2.856	1.939	2.598	86.508	23.545	27.2	100	100553	0	0	98435	2118	0	0
2	0.246	1.532	2.221	2.843	1.945	2.584	77.372	22.196	28.7	100	87594	0	0	85764	1830	0	0
3	0.247	1.528	2.220	3.003	1.944	2.744	91.577	23.422	25.6	100	106298	0	0	104076	2222	0	0
4	0.241	1.520	2.162	3.455	1.887	3.196	104.934	24.720	23.6	100	125753	0	0	123310	2443	0	0
5	0.251	0.536	0.573	0.577	0.318	0.325	0.001	0.000	0.0	0	13	0	13	0	0	0	0

Figure 4 –L4S Traffic Results at end-to-end network packet level with no bottleneck network segment (L4S application benchmarking)

In another case (Figure 5 and Figure 6), microbursts of a short duration but with a high amplitude cause multiple L4S packets to queue before arriving at the DOCSIS network. Since the previous network does not support L4S, every burst effect will be reflected as queue-building arrival process to the access network. The DOCSIS network will mark bursty L4S traffic with CE and sanction a set of packets in the bursty arrival. As shown in the figures, this creates a disruption in the L4S traffic's throughput but the 99.9 percentile of traffic latency is still less than 3 ms in the downstream and the application round-trip time (RTT) smooths the peaks within 100 ms.

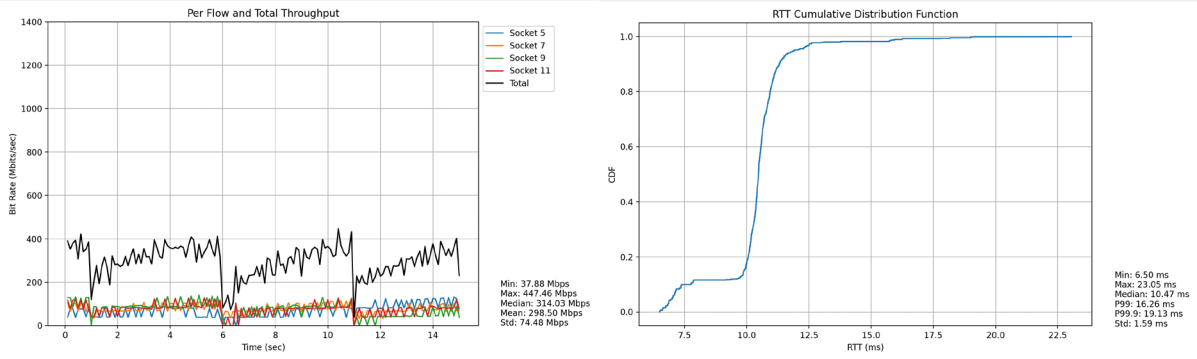
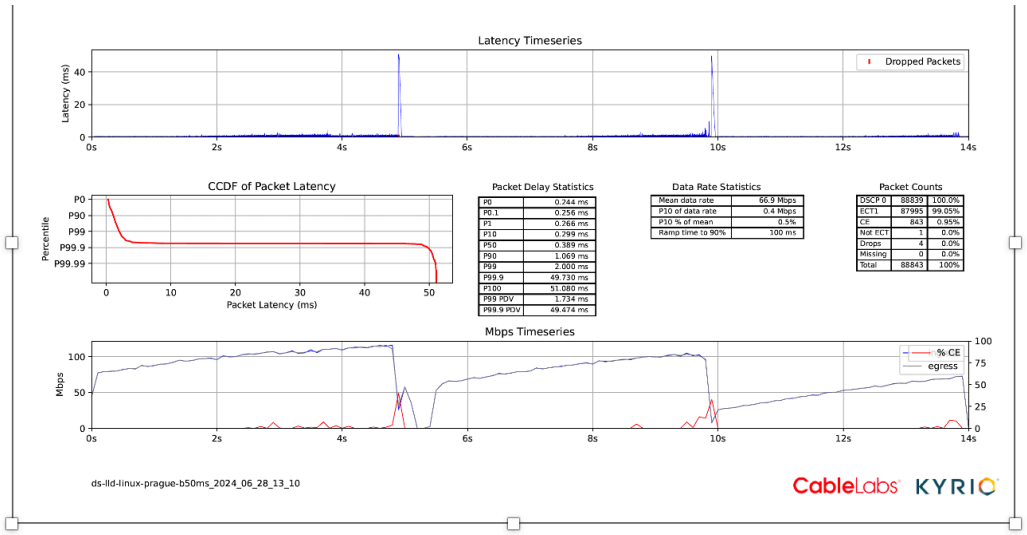
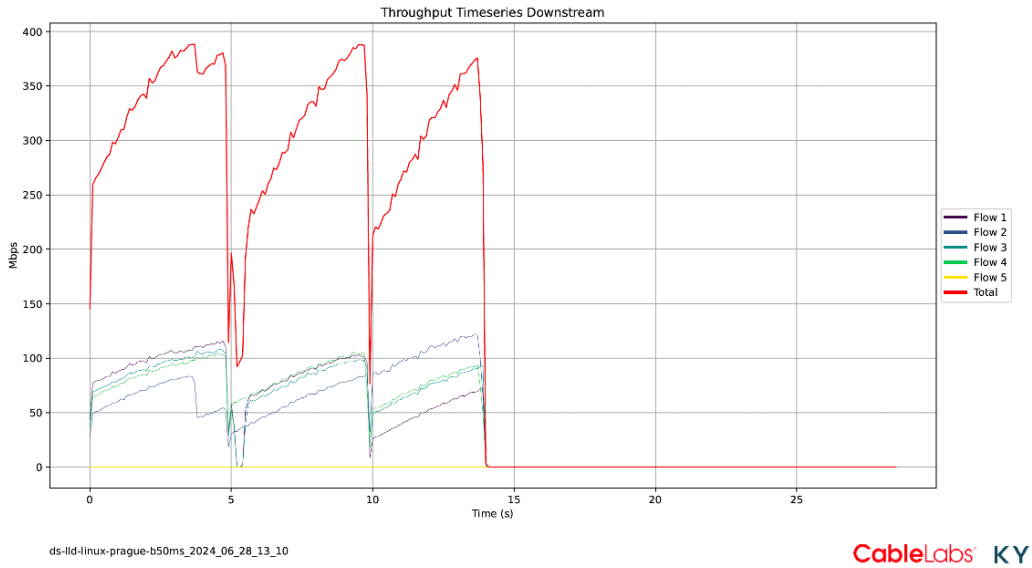


Figure 5 – L4S Traffic Results at Application Layer (100ms measurement interval) with microbursts in the initial network segment (single L4S traffic)

The top graph in Figure 6 shows that the majority of CE markings happened during the microbursts in the network segment outside of the access network. The sensitivity of the reaction to queue-building traffic levels can be adjusted via the LLD parameters. The IETF and CableLabs L4S interops unify testing among industry, academia and open-source organizations. Guidelines on the test cases and parameters can be found at the standards' websites. The collaboration enables all parties, including network, service, content, and OS providers, as well as researchers, to test and develop the L4S implementations. Although other network technologies have started to integrate L4S, not all network segments will support L4S in the near future. It is important to build a knowledge base for end-to-end system behavior including transient/peering and home [2] networks. These tests use the latency analysis tools used by CableLabs and IETF interops and provide full transparency.



Aggregate_thruput_downstream.pdf



summary_table_downstream

Flow Number	Packet Delay P0 (ms)	Packet Delay P90 (ms)	Packet Delay P99 (ms)	Packet Delay P99.9 (ms)	PDV P99 (ms)	PDV P99.9 (ms)	Mean data rate (Mbps)	P10 of data rate (Mbps)	P10 % of mean	Ramp time to 90% Tput (ms)	Num Packets	Num NotECT	Num ECT0	Num ECT1	Num CE	Dropped Packets	Missing Packets
1	0.244	1.069	2.000	49.730	1.734	49.474	66.891	0.351	0.5	100	88843	1	0	87995	843	4	0
2	0.246	1.001	2.036	49.081	1.770	48.825	69.607	0.511	0.7	1700	84951	0	0	84148	802	1	0
3	0.248	1.042	2.042	49.837	1.775	49.382	71.372	0.410	0.6	200	91345	1	0	90461	878	5	0
4	0.245	1.038	1.977	49.803	1.712	49.547	75.191	0.770	1.0	800	94027	0	0	93190	835	2	0
5	0.261	0.536	0.553	0.555	0.290	0.294	0.001	0.000	0.0	0	17	4	13	0	0	0	0

Figure 6 – L4S Traffic Results at end-to-end network packet level with microbursts in the initial network segment (single L4S traffic)

In the following, we show examples where multiple L4S and classic service flows (SFs) co-exist in the system. Previous network segment's latency contribution can be modeled based on the available data in the literature. The benchmarking case in Figure 4 and Figure 5 provides a base for the L4S application and LLD network interaction. Figure 7 displays the histograms reported by the CMTS while Figure 8 displays packet capture analysis with throughput and latency statistics. DOCSIS latency histograms enable ISPs to measure latency at the service flow level while congestion metrics provide additional visibility to the performance of the L4S services. The consistent low latency of L4S traffic can be confirmed in these figures. The fairness among L4S and classic flows in terms of throughput is captured in Figure 8. The 99.9 percentile latency for L4S flows is still less than 3 ms in the presence of shared medium and device. The increased CE markings help the L4S traffic react to the busier medium without increasing its latency.

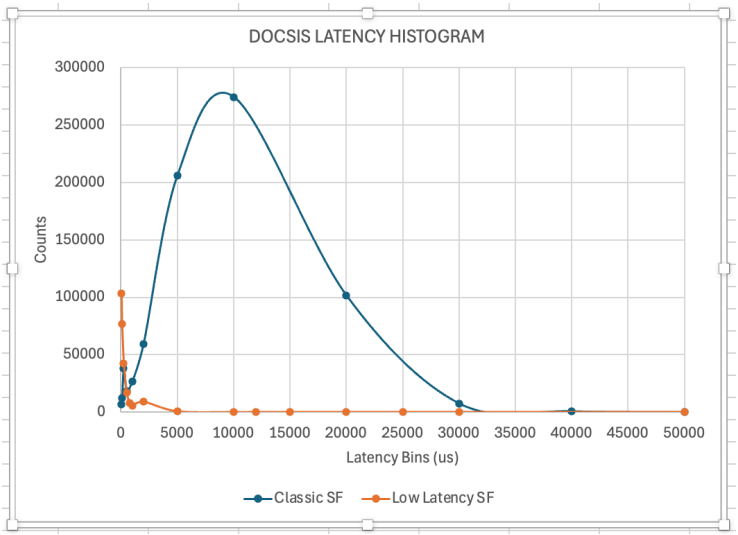


Figure 7 – Latency Histogram from DOCSIS Management Information Base (MIBs): multiple L4S and classic traffic with no additonal latency issues

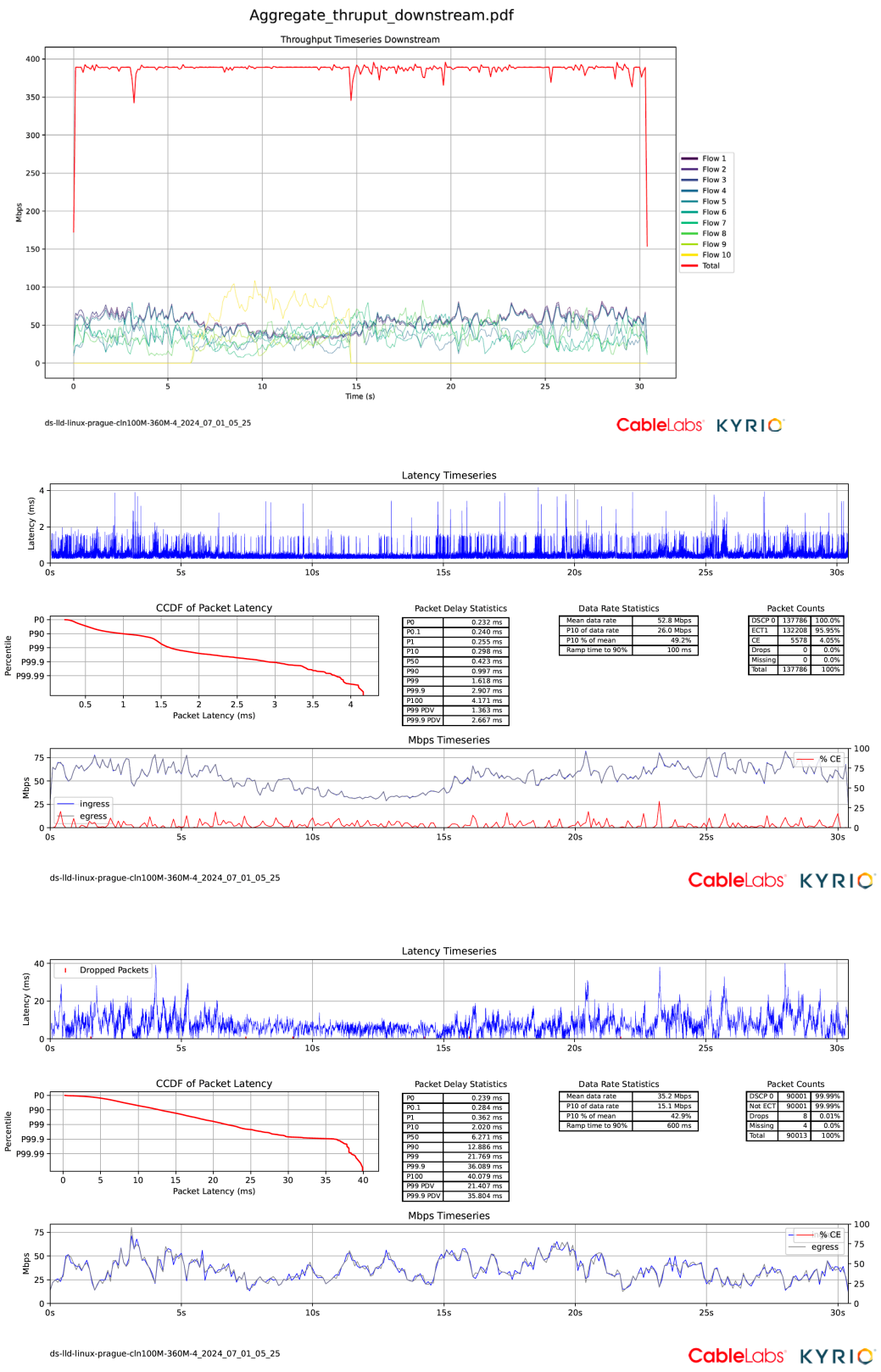


Figure 8 – Latency and Throughput analysis of L4S and classic flows: multiple L4S (middle) and classic traffic (bottom) with no additonal latency issues

In the next scenario (Figure 9 and Figure 10), microbursts again cause the L4S traffic to scale back due to the higher number of CE markings in the DOCSIS network where queue protection detects the arrival traffic as queue building. The L4S traffic reacts fast to the CE markings from the CMTS and keeps the latency values consistently low. When the system is utilized more, the impact of the burst on the single L4S flow is less compared to the case illustrated in Figure 6. Although it may sound counterintuitive, this is the expected behavior of the impact of microbursts when more favorable conditions change abruptly. For most real-time interactive and immersive applications, the jitter is more crucial than the median latency. Therefore, it is important to test microbursts that may happen in the home, serving groups of networks, core and outside of the ISP network. Microbursts are harder to detect, but in this case, since we can compare the typical behavior of the LLD network and LL SF within the aggregate SF, the DOCSIS CE markings and histogram changes can be used to detect the anomalies in the previous network segment. Additional knowledge on other DOCSIS metrics such as PHY layer statistics can be used to isolate the issues from different network segments. Sometimes, the microburst may not be rare but detecting them with measurement tools may be rare due to the practical issues. In clean use cases, we can estimate the bursty latency contributor of a previous segment from a well-known L4S application traffic as shown in this case. The knowledge base helps to build and confirm prediction and causal learning models and select optimal sets of measurements.

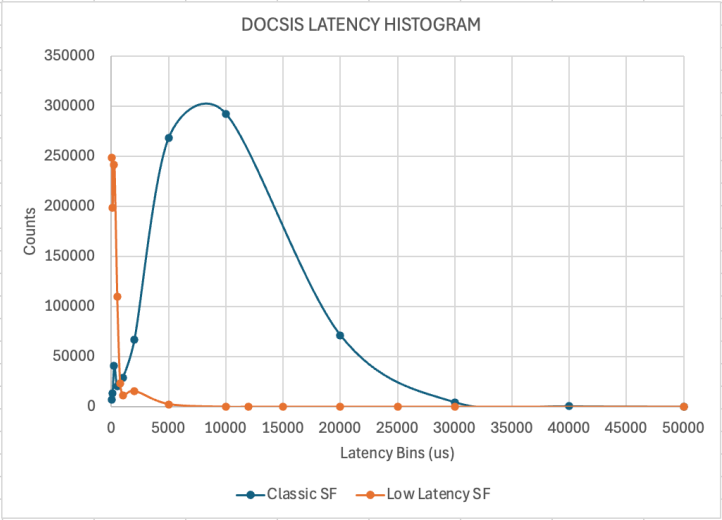


Figure 9 – Latency Histogram from DOCSIS MIBs: multiple L4S and classic traffic with microbursts in the initial network segment

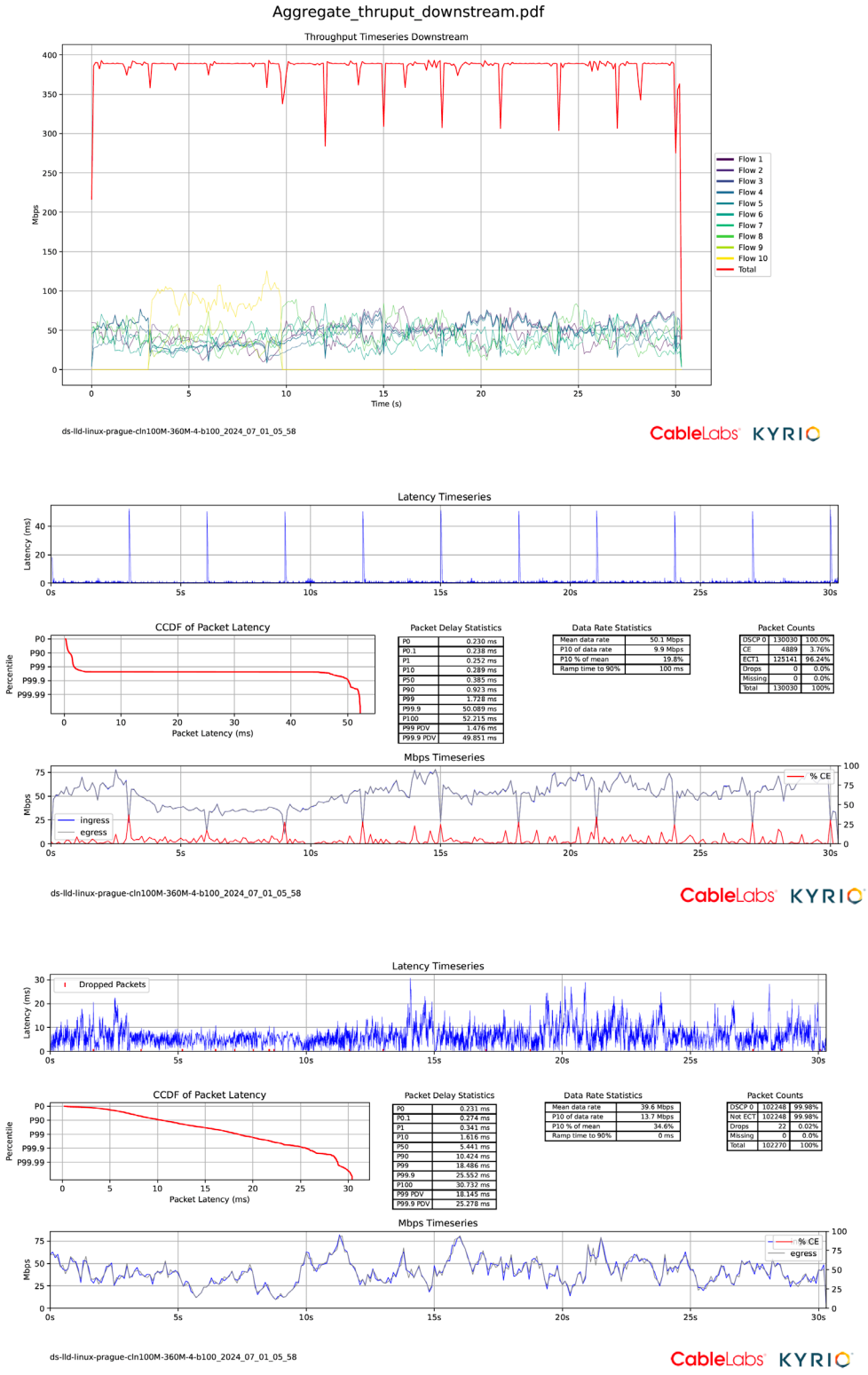


Figure 10 – Latency and Throughput analysis of L4S and classic flows: multiple L4S (middle) and classic traffic (bottom) with microbursts in the initial network segment

When utilization is high, a large number of samples for a bounded system can be modeled with a Gaussian distribution as shown in Figure 11 and Figure 12, although we cannot exclude heavy tail behavior in a more general case. In this case, DOCSIS latency is measured, while backhaul network latency is fit using a Gaussian distribution with a mean of 10 ms and a standard deviation of 2 ms. The average and median end-to-end latency for the L4s traffic are increased, but 99 and 99.9 packet delay variation (PDV) values show consistently low values compared to those of the classic traffic. This is one of the examples where a simple tandem network model fits well. The total mean latency is the sum of the means of the individual segments, and the total variance is the sum of the variances of the individual segments. The analysis becomes more complex with L4S traffic due to the fact that bottleneck in the previous network segment can affect the arrival pattern for the next segment during this time, as shown in the previous example. Since the QoE requirements may have 99 and higher percentile latency targets, it is crucial to analyze the network bursts. Our aim is to analyze both slowly varying disturbances and high frequency variances separately and through probabilistic superposition over different time segments and network conditions.

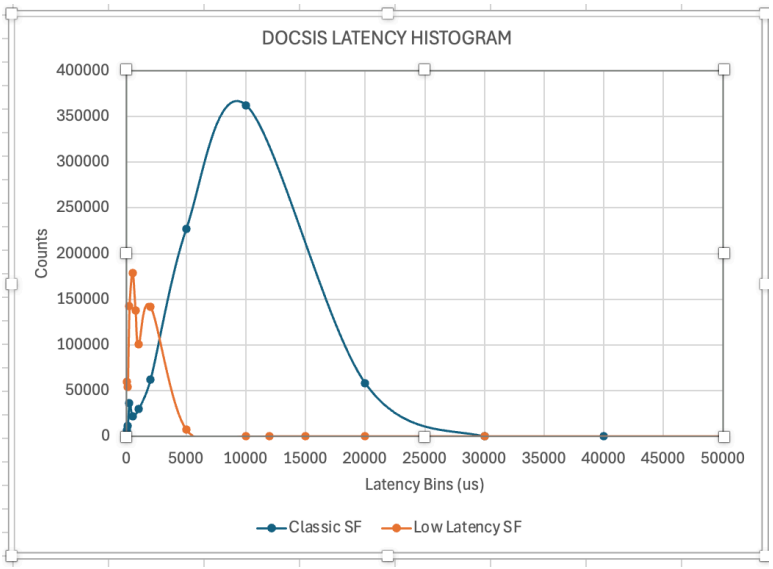
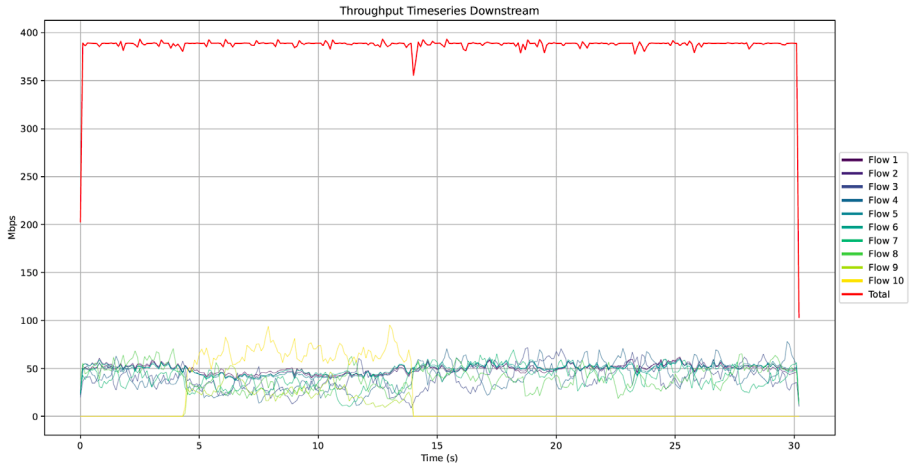


Figure 11 – Latency Histogram from DOCSIS MIBs: multiple L4S and classic traffic with additional Gaussian distributed latency in the previous network segment

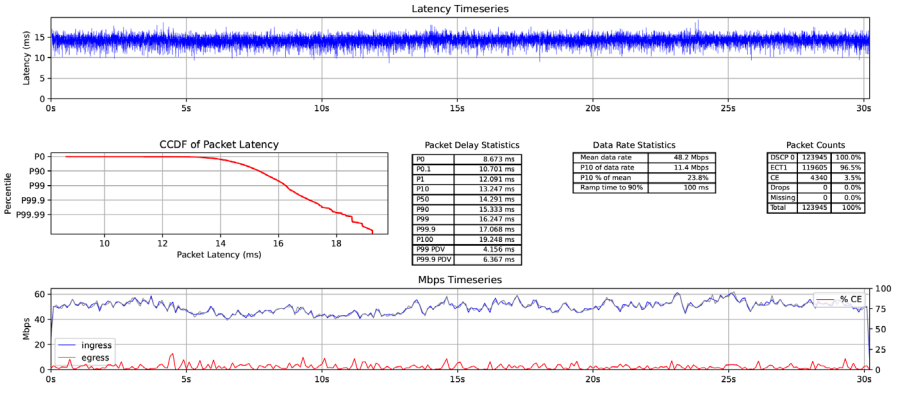
Aggregate_thruput_downstream.pdf



ds-lld-linux-prague-cln100M-360M-4-gauss10-2_2024_07_01_06_11

CableLabs KYRIO

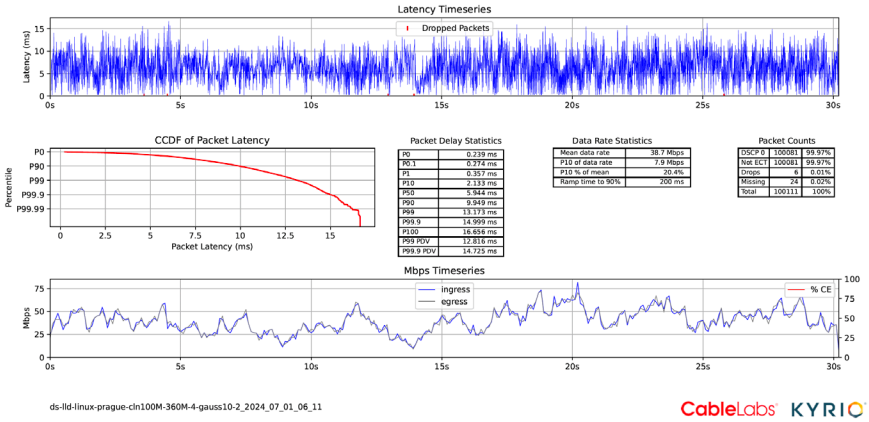
1_downstream.pdf TCP_src_71.85.92.86_37172_to_dest_71.85.92.211_5206



ds-lld-linux-prague-cln100M-360M-4-gauss10-2_2024_07_01_06_11

CableLabs KYRIO

3_downstream.pdf TCP_src_71.85.92.82_59693_to_dest_71.85.92.210_5207



ds-lld-linux-prague-cln100M-360M-4-gauss10-2_2024_07_01_06_11

CableLabs KYRIO

Figure 12 – Latency and Throughput analysis of L4S and classic flows: multiple L4S (middle) and classic traffic (bottom) with additional Gaussian distributed latency in the previous network segment

Our last scenario is modeled with a uniform RTT that stays stable during the L4S session while the impact of lower utilization on the latency can be observed (Figure 13). The distribution of end-to-end latency is more uniform in this case and can be associated with the longer distance to the application server. This is a good starting use case to test the LLD coupling feature to assess the throughput fairness between classic and L4S service flows towards the same cable modem. As seen in Figure 14, this fairness level is kept throughout the session. Some of the flows are generated using the CableLabs interops script with high varying loads within sub-ms time frames. As in the previous case, the superposition of multiple latency and loading conditions help to build the knowledge base for causal inference.

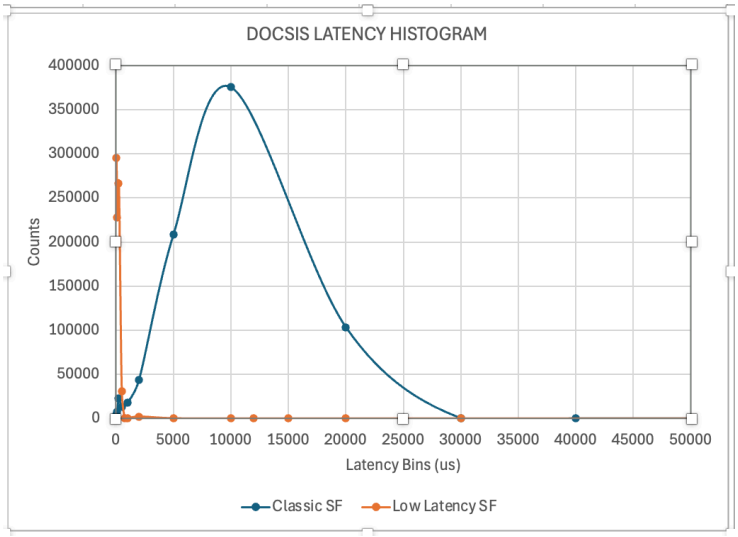


Figure 13 – Latency Histogram from DOCSIS MIBs: multiple L4S and classic traffic with additional Uniformly distributed latency in the previous network segment

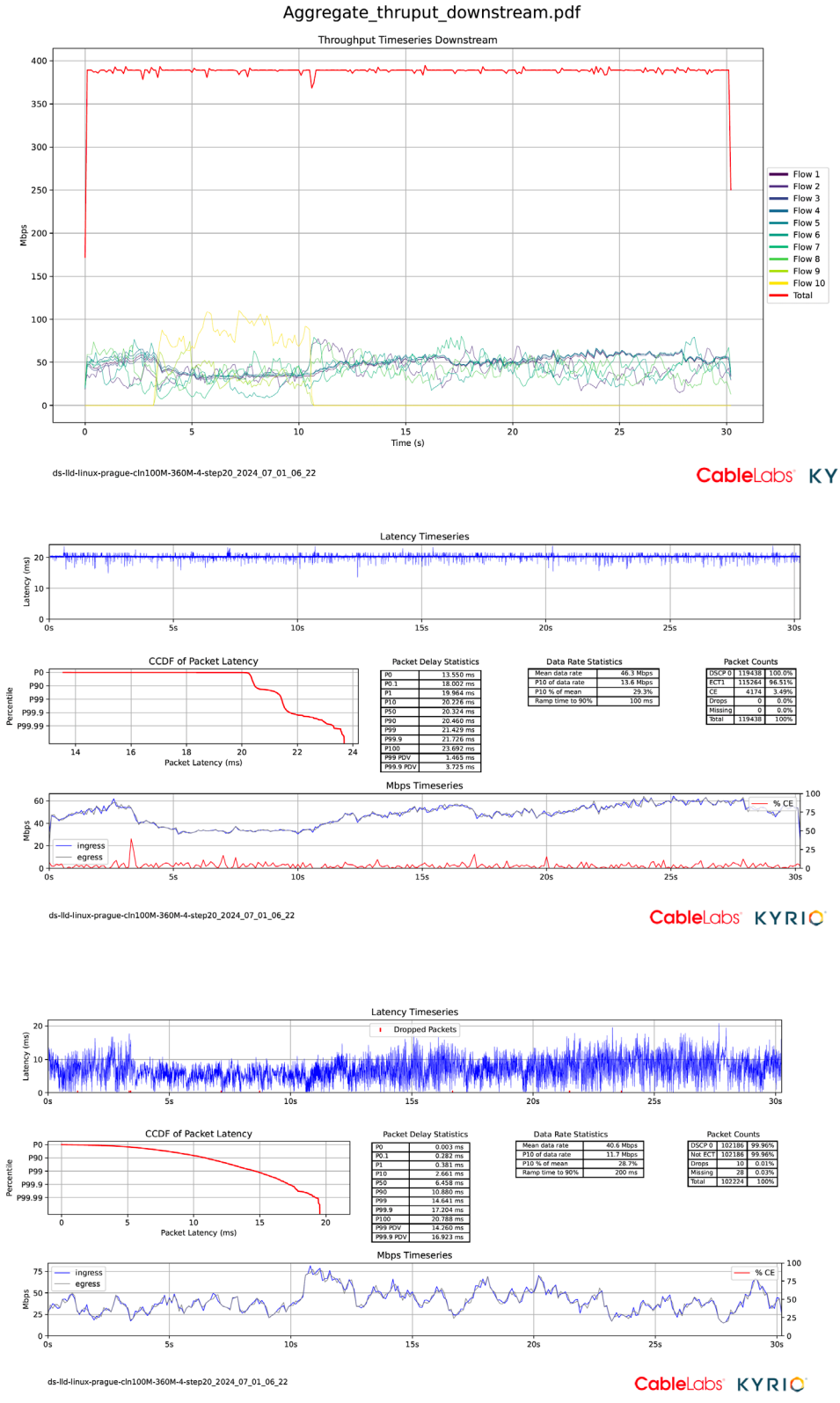


Figure 14 – Latency Histogram from DOCSIS MIBs: multiple L4S and classic traffic with additional Uniformly distributed latency in the previous network segment

3. Latency Prediction and Causal Inference

While we showed examples of different sources of latency contributions in the uncontrolled network segment, in real scenarios, a mix of these use cases can be the determining factor of the L4S session's quality. We used these cases to analyze the behavior of L4S traffic under different uncontrolled impairment conditions. We then used prediction and causal machine learning models to predict the latency behavior of service flows and to understand the conditions that cause latency variation and how system variables affect the service quality.

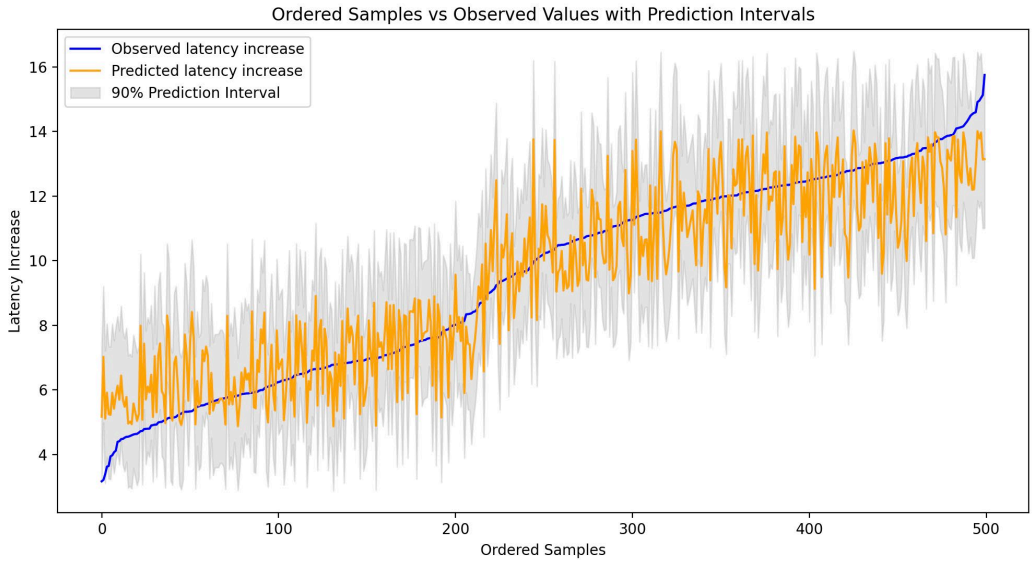


Figure 15 – Forest Regression Model

Even though the system used distinct latency use cases, the prediction (Figure 15) shows that at the very low and very high utilization and latency cases, the prediction is weak. We can expect lower prediction in the actual deployments. However, we can improve the models by using knowledge base based on the detailed analysis of the queueing schemes and advanced passive latency measurement tools. Furthermore, this knowledge base can be applied to causal analysis and inference models.

Regression analysis is a common method used to explore relationships between variables. Causal Forest models, on the other hand, are designed to understand the causal effects of different factors. We applied our data to the Causal Forest model for our research. Forest causal models are a powerful tool used in data science to understand how different factors (called treatments) affect outcomes, and to explore these effects in more detail than traditional methods. For example, LLD optimization parameter can be a treatment that affects latency outcome. Forest causal models, such as Causal Forests estimate the Conditional Average Treatment Effect (CATE) that shows the cause-effect relation. In our case, we found a strong CATE value (~-20) between the additional RTT in the uncontrolled network segment and quality of the end-to-end system.

However, causal models are still at an early stage although the term was first used four decades ago. Most of the current models are predictive tools that cannot reason the observed behavior or relation. Furthermore, both predictive and causal models have limitations in mitigating data errors, especially systematic errors.

Systematic errors in data can lead to flawed models and incorrect conclusions, regardless of whether the AI system is designed for predictive or causal purposes. Using causal AI for decision making without mitigation of systematic errors can cause more harm. Human experts, equipped with a deep knowledge base and practical experience, can identify and correct these errors through methods like lab tests and troubleshooting in specific fields such as communications systems. AI, however, requires specific training to detect such errors. This can be achieved through the incorporation of robust statistical methods, anomaly detection algorithms, and domain-specific rules that mimic expert knowledge. Despite advances in AI, the technology is still developing, and human expertise remains essential for the current solutions. Knowledgeable human experts are crucial for validating data and the results produced by AI systems, ensuring that causal inferences and predictions are accurate and reliable. This mutual relationship between AI and human expertise helps to safeguard against errors and enhances the overall trustworthiness of current AI applications.

In the following, we analyze an active latency measurement test data, which is not optimized for different speed tier rates. This data includes generated traffic latency to detect additional round trip times due to uncontrolled network segments, misconfigurations, worst case utilization scenarios, and link and device issues. Therefore, it shows high latency values but is not always accurate for high rates. As ISPs increase the rates both in downstream and upstream, the measurement requirements change as well. When these techniques are not optimized for a different network, device and service conditions, the results can be misleading, especially when a systematic error causes a strong correlation among the system variables. As shown in Table 1, the mean and median values for higher speed tier rates is almost half of the values of the lower speed tier rates. Without a solid knowledge base or tools for the machine learning model to detect systematic errors, one can misinterpret the statistics as the higher rate helps to reduce the latency measured by this test (Figure 16). However, this test measures the latency under load for a given time and the probability of having bursts of high queue latency. Since this specific system uses the same queueing model and parameters for all the rates, the results should converge. There must be no significant impact of the rate on the mean and median downstream (DS) latency under load (LUL) test results. However, when we compute the correlation values among the system variables, we find a correlation value of -0.53 between the rate and median DS LUL values. A knowledge base system can easily detect the error and further analysis can lead to changes to the test parameters to provide accurate measurements. Without providing the interworking of the network functionalities and the test system, the machine learning model may not detect the error. For this example, the variance at each rate provides a verification point as LUL values should converge if path conditions remain the same.

Table 1– DS LUL Statistics per Speed Tier Rate

Rate (Mbps)	Mean (ms)	Median (ms)
50	112.64	110.49
100	106.98	109.55
300	106.54	107.57
500	91.71	93.44
1000	64.92	67.49

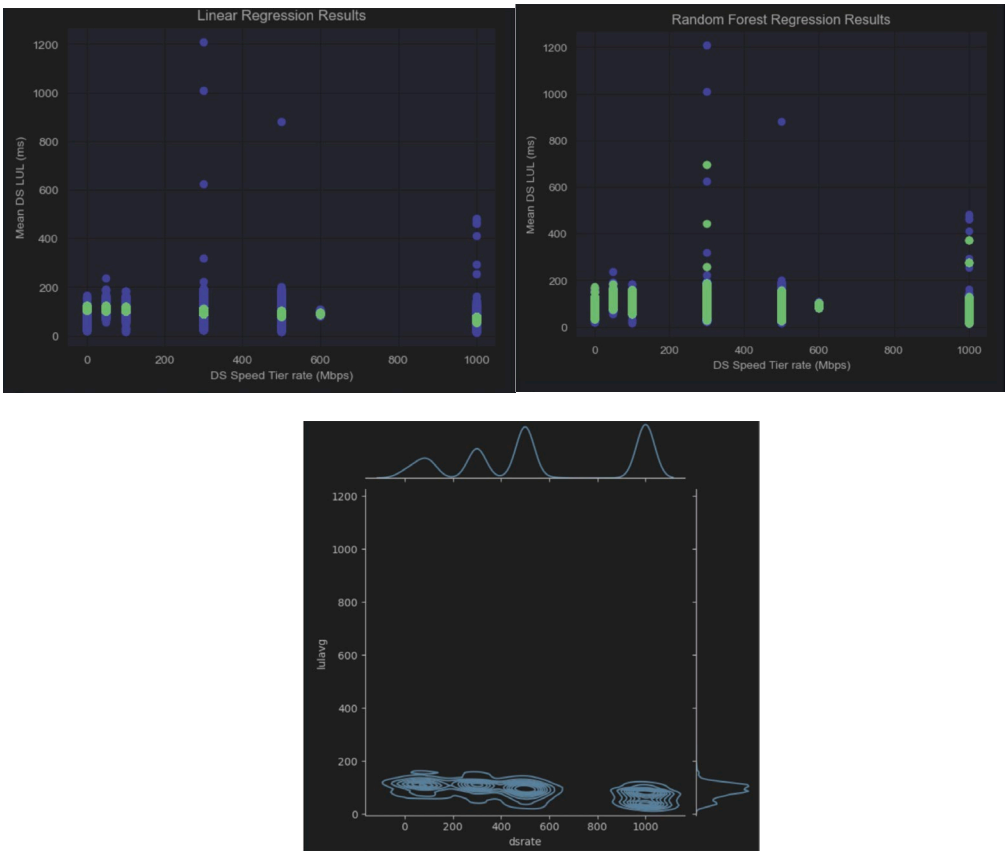


Figure 16 – Top: Linear and Forest Regression Models of the DS LUL data (without preprocessing), Bottom: Kernel density estimation with rate (top) and LUL (right) disributions

Although the current models can benefit from human expertise, the knowledge base can be integrated into a self-correcting model with the advances in the machine learning techniques, generative AI and methods that facilitate the input of a specific expertise into generic platforms (Figure 17). An example of this is reinforcement learning, where an agent learns to make decisions by interacting with an environment. It learns through trial and error, receiving feedback in the form of rewards or penalties based on its actions.

RL algorithms aim to maximize cumulative rewards over time. Observations and context can define the state representation that captures relevant variables and features from the environment, along with historical data. The RL agent then decides actions based on the current state representation with policy optimization (e.g., Q-learning, policy gradients). Causal forest predictions can first be evaluated by experts through an effective feedback loop in a continuous development model, which leads to conditional generative models. The comparison of predicted outcomes with actual outcomes is used to detect discrepancies or errors for adaptive learning and self-correction mechanisms.

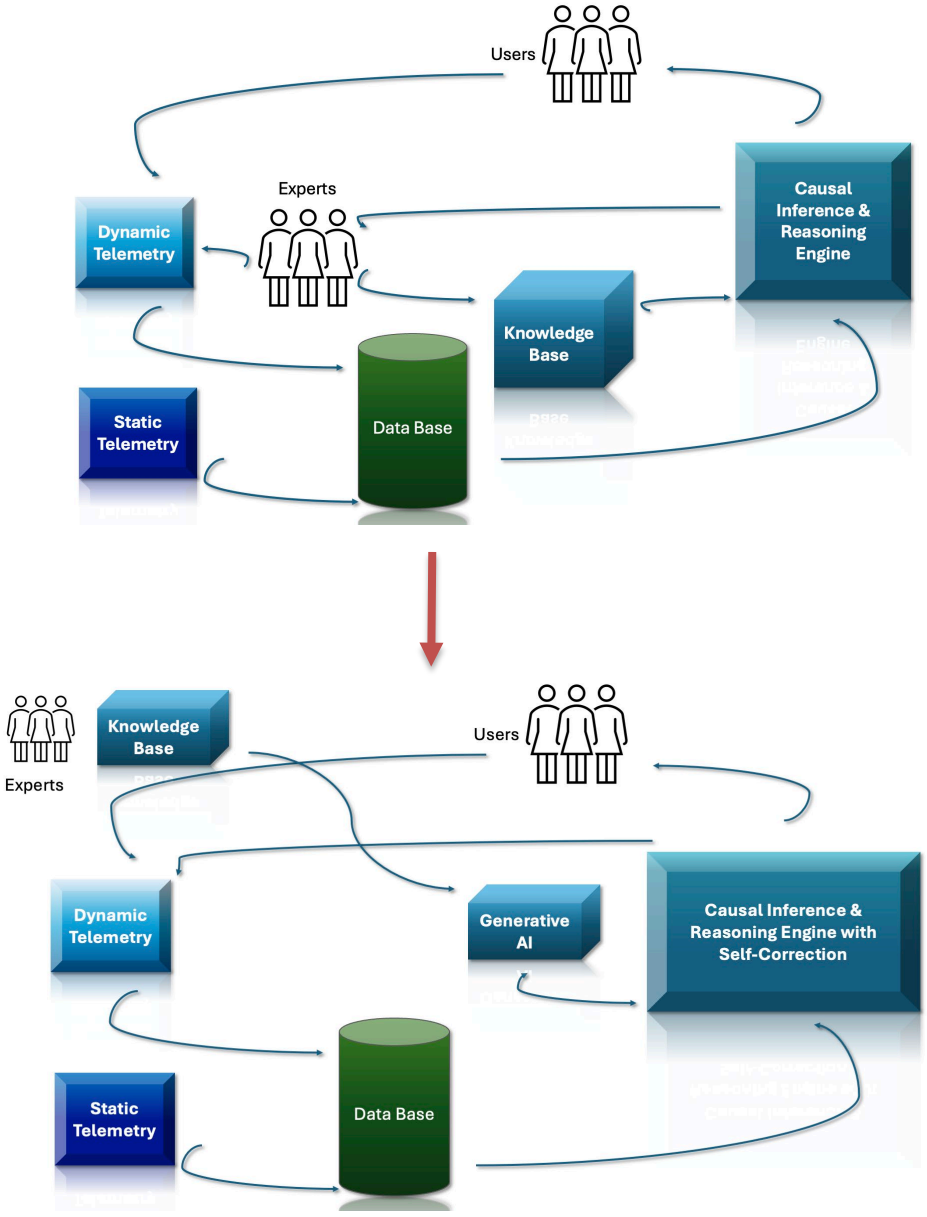


Figure 17 – Towards (almost) autonomous machine learning models

4. Conclusion

In this paper, we analyzed coupled latency variations between different network segments. We discussed detailed tests carried for IETF L4S traffic over Low Latency DOCSIS and internet networks that do not

support L4S. We then used the knowledge base to find predictions and causal inference based on confirmed data input. Our findings with another set of test data confirmed that without human expertise, current AI models may mislead the operators when the data has systematic errors. This led us to provide a two-step approach to use causal inference models, integrated with human expertise and generative AI.

Abbreviations

AI	artificial intelligence
bps	bits per second
CE	congestion experienced
CM	cable modem
CMTS	cable modem terminal system
CATE	conditional average treatment effect
DOCSIS	Data Over Cable Service Interface Specifications
DS	downstream
ECN	explicit congestion notification
ECT	ECN capable transport
IETF	Internet Engineering Task Force
ISP	Internet Service Provider
L4S	low latency low loss scalable traffic
LLD	low latency DOCSIS
LUL	latency under load
MIB	management information base
PDV	packet delay variation
QoS	quality of service
RTT	round-trip time
SF	service flow
SCTE	Society of Cable Telecommunications Engineers
TCP	transmission control protocol

Bibliography & References

1. *Breaking the Barriers: Abstracting Traffic Management for Superior Quality of Experience in Multi-Technology Networks*, Sebnem Ozer, Ramneek Bali, Kamran Yousuf & Moutaz Elkaissi, SCTE 2023
2. *Wi-Fi Access Latency Characterization*, Lei Zhou et. al., SCTE 2024
3. Athey, S., & Imbens, G. W. (2016). Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113(27), 7353-7360.
4. Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, 113(523), 1228-1242.

Acknowledgments

The authors would like to thank and acknowledge Daniel Lee, Timothy Welch and Christian Mellado for their contributions to the LLD lab and tests.