Measuring Low Latency at Scale: A Field Study of L4S in Residential Broadband

Ayoub Ben-Ameur[1], Francesco Bronzino[1,2,3], Paul Schmitt[1,4], and Nick Feamster[1,5]

[1] NetMicroscope Inc., Chicago, IL 60637, USA firstname@netmicroscope.com
[2] École Normale Supérieure de Lyon, Lyon 69007, France
[3] Institut Universitaire de France, Paris 75005, France
[4] California Polytechnic State University, San Luis Obispo, CA 93407, USA
[5] University of Chicago, Chicago, IL 60637, USA

Abstract. The Low Latency, Low Loss, Scalable Throughput (L4S) architecture promises to reduce queuing delay while sustaining high throughput. Prior work has largely evaluated L4S in synthetic environments or controlled testbeds, leaving its real-world performance underexplored. In this study, we measure L4S performance specifically on **Apple services delivered over Comcast residential networks**. We deploy 83 Raspberry Pi devices across Comcast subscriber households and conduct over 120000 controlled experiments comparing L4S to traditional congestion control. Our results show that L4S reduces tail latency by up to 25% for interactive applications and for bulk downloads from Apple's CDN, while providing minimal gains for iCloud. Gains are most pronounced during peak usage hours when networks are congested, highlighting the situational benefit of L4S in a single ISP ecosystem.

1 Introduction

The Internet is undergoing a fundamental shift in performance priorities. For decades, throughput was the primary bottleneck, driving innovations in link capacity, backbone infrastructure, and content distribution. Today, with widespread gigabit access and abundant bandwidth, latency has emerged as the critical performance metric. Modern interactive applications (*e.g.*, video conferencing, cloud gaming, remote collaboration, and real-time communication) are increasingly constrained not by available bandwidth, but by end-to-end delay and its variability.

This shift poses a fundamental challenge for Internet congestion control, which has historically optimized for throughput and fairness. Traditional Active Queue Management (AQM) schemes and congestion signaling techniques struggle to maintain low queuing delay under load, often forcing a tradeoff between latency and link utilization. The result is persistent bufferbloat [6] and unpredictable latency, even on high-capacity access links.

To address this, the Internet Engineering Task Force (IETF) has proposed the Low Latency, Low Loss, Scalable Throughput (L4S) architecture [4,12], which reimagines congestion control by enabling ultra-low queuing delay through scalable Explicit Congestion Notification (ECN) marking and new transport behaviors. L4S introduces dual-queue AQMs (such as DualPI2 [13]) and transport algorithms that respond more aggressively to ECN signals, aiming to keep queues short while maximizing link utilization.

While L4S has been evaluated extensively in controlled testbeds and simulations [3,9,14,7], its real-world performance remains largely unmeasured. This gap is particularly critical as L4S transitions from research prototype to production deployment. Recently, several major Internet players have begun enabling L4S: Apple has integrated L4S support into iOS and macOS [1] for Facetime, iCloud, and CDN downloads; NVIDIA has deployed L4S in its GeForce NOW cloud gaming platform [11]; and Comcast has started provisioning residential customers with L4S-capable dual-queue routers [5]. This emerging deployment creates an unprecedented opportunity to measure L4S performance at scale in production networks, under real traffic conditions, with actual applications—measurements that are impossible to replicate in controlled environments.

In this paper, we present the first large-scale, in situ measurement study of L4S in a single production network. We deploy 83 Raspberry Pi measurement nodes across Comcast residential broadband networks in the United States, conducting over 120000 controlled experiments comparing L4S-enabled Apple services (FaceTime, iCloud, and Apple CDN downloads) against their traditional congestion control baselines. Our measurement methodology combines active probing, packet-level instrumentation, and application-layer metrics to quantify L4S impact on latency, throughput, packet loss, and application responsiveness within Comcast s access network and under its real-world operating conditions.

Our key contributions are as follows:

- We design and deploy a large-scale measurement infrastructure spanning 83 residential networks to measure L4S performance in production, conducting over 60,000 controlled experiments across diverse ISPs, access technologies, and network conditions.
- We demonstrate that L4S benefits are highly context-dependent: tail latency reductions of up to 25% for interactive applications occur primarily during peak congestion hours, while benefits are minimal during off-peak periods and for certain application types (e.g., iCloud sync).
- We identify both the benefits and limitations of L4S in real-world deployments, including deployment challenges and coexistence concerns.

Our findings reveal that L4S delivers measurable latency improvements in production, but its benefits are concentrated in specific contexts (*e.g.*, congested networks, peak hours, and latency-sensitive applications). These results provide the first empirical evidence of L4S effectiveness at scale and highlight both the promise and practical limitations of deploying low-latency congestion control in the Internet.

2 Related Work

While L4S has received considerable attention in both academic and standardization communities, most evaluations to date have been confined to testbeds or simulations. This study complements prior work by providing an empirical analysis of L4S performance in production networks with real-world traffic.

Controlled testbed evaluations. Early empirical validation of L4S [3] focused on fairness between scalable and classic congestion controls under controlled experiments, demonstrating substantial queuing delay reductions while identifying coexistence challenges when flows share bottlenecks. Graff *et al.* [7] evaluate L4S in a custom platform that imitates cloud gaming, implementing the Self-Clocked Rate Adaptation for Multimedia (SCReAM) algorithm under emulated cellular conditions. Their work examines fairness and Quality of Service (QoS) for synthetic traffic, comparing L4S with class-based queuing. Monteiro *et al.* [8] examine L4S in a private 5G industrial setting for real-time video streaming, measuring latency, throughput, and video quality. They show significant queuing delay reductions under certain traffic loads within their controlled network. While these studies provide valuable insights into L4S behavior under specific conditions, they rely on synthetic traffic patterns and controlled environments.

Network-specific contexts. Srivastava *et al.* [14] investigate low-latency congestion control protocols—TCP BBR and TCP Prague—over mmWave links, which experience frequent capacity drops due to blockage and rapid variations. Their results show that while these protocols reduce queueing delay under many conditions, fairness issues emerge (some flows starve), and frequent capacity disruption limits achievable latency improvements. This work highlights how specific link characteristics constrain L4S benefits, though in controlled rather than production settings.

Architectural and instrumentation perspectives. Complementary work has explored L4S from architectural and measurement perspectives. Szilveszter *et al.* [9] challenge the tight coupling between L4S and specific scalable congestion control algorithms, introducing a scheduler that delivers low-latency service regardless of sender congestion control. Their primarily algorithmic work, evaluated in controlled settings, demonstrates the feasibility of decoupling architectural benefits from end-host adoption. Nguyen *et al.* [10] leverage programmable data planes to observe and validate L4S flow behavior in fine-grained detail, building a P4-based framework with in-band network telemetry that captures per-packet latency and congestion signals. Their focus on data plane instrumentation provides valuable tools for debugging and validation.

Our contribution. In contrast to these controlled and instrumentation-focused studies, we provide large-scale, in-situ measurements of L4S across residential broadband deployments. We quantify end-to-end performance that end-users experience with L4S-enabled commercial Apple services over Comcast networks, using live traffic from geographically distributed devices. This empirical analysis reveals how L4S performs under real-world conditions, including varying congestion levels, diverse service types, and partial deployment scenarios.

3 Methodology

We design a measurement campaign to assess L4S performance in real-world environments across representative application scenarios. Our study systematically compares traffic using L4S-enabled congestion control against traditional queueing. We focus on three categories of Apple services that span latency-sensitive and throughput-oriented applications: (i) Apple CDN downloads, (ii) iCloud, and (iii) FaceTime. These services were selected because Apple has recently enabled native L4S support across its application stack and content delivery infrastructure, providing a rare opportunity to evaluate L4S in production conditions with unmodified end systems. Moreover, they collectively represent distinct transport behaviors: Apple CDN downloads produce long-lived, high-throughput transfers that reveal how L4S handles sustained congestion; iCloud downloads exhibit short, bursty synchronization flows typical of background traffic; and FaceTime calls generate continuous, interactive streams that stress low-latency performance.

3.1 Experimental Design

For each service, we define reproducible test procedures executed periodically under two network configurations:

1. **L4S-enabled:** ECN with ECT(1) marking and Dual Queue support on the access link, enabling scalable congestion control.
2. **Non-L4S:** Conventional congestion control without ECN or with classic ECN (ECT(0)).

We employ a paired testing approach where L4S and non-L4S measurements execute in immediate succession (*i.e.*, within 60 seconds) to minimize temporal variability. This design controls for time-of-day effects, transient congestion, and routing changes that could confound comparisons. Each paired test alternates the order (L4S-first vs. non-L4S-first) to account for potential ordering effects.

3.2 Measurement Procedures

We instrument each service type to collect network and application-layer metrics:

Apple CDN download tests. We download large media files (480 MB) hosted on Apple CDN infrastructure using HTTP/2 over TCP or HTTP/3 over QUIC. We measure time-to-first-byte (TTFB), total transfer time, achieved throughput, and path-level RTT. This workload represents high-throughput, non-interactive usage.

iCloud tests. We trigger downloads of 500 MB files through a headless iCloud client session, ensuring repeatable transfer sizes. We log application-layer throughput and TCP RTT. Network-layer measurements include congestion window evolution and ECN activity (ECT/CE markings).

FaceTime tests. We establish automated two-party video calls and use passive packet capture (tcpdump) to measure call establishment time, ECN markings,

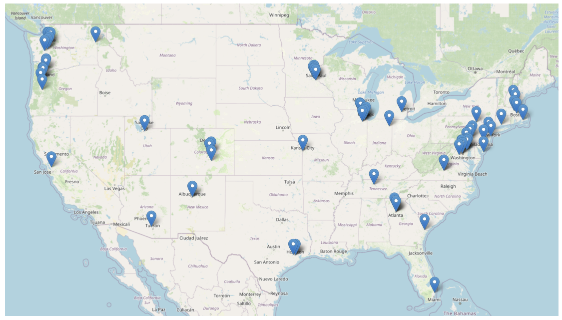


Fig. 1: Geographical distribution of deployed devices across the United States

and packet loss. These metrics quantify the impact of queue management and congestion control on real-time interactive performance.

3.3 Measurement Infrastructure

We deploy a distributed measurement infrastructure consisting of 83 Raspberry Pi devices in volunteer residential homes across the United States (Figure 1). We manage the fleet remotely using the openBalena [2] platform, enabling over-the-air software updates, configuration management, and centralized logging without requiring on-site intervention.

All volunteer homes subscribe to Comcast Xfinity residential broadband service with advertised speeds up to 1 Gbps downstream and 300 Mbps upstream. Critically, Comcast has deployed routers with dual-queue AQM support in these homes, satisfying the infrastructure requirements for L4S operation. Each device connects via Ethernet to the home router, ensuring stable connectivity and minimizing wireless interference in measurements.

Devices are geographically distributed across multiple US regions to capture diverse network conditions including varying ISP peering relationships, regional congestion patterns, and path characteristics. Each device executes the measurement procedures described in Section 3, alternating between L4S-enabled and non-L4S configurations. Measurements run autonomously 24/7, capturing diurnal traffic patterns and congestion dynamics.

3.4 Final Dataset

Tests execute every 3 hours across a 48-week measurement period, yielding approximately 2688 paired measurements per service type per deployment site.[6] We schedule tests to capture diurnal patterns. Each test cycle generates structured logs containing: timestamps, ECN counters, flow-level statistics, and application

[6] Data collection was progressively activated across devices and services; consequently, not all nodes or applications contributed measurements during the exact same time intervals.

Table 1: Summary of the dataset.

Collection period	Nov. 1st 2024 – Sep. 30th 2025
Country	United States of America
Number of homes	83
Experiments per service per day	8
Total experiments per service	Apple CDN (50000); iCloud (35000); FaceTime (35000)
Collected metrics	Latency (ms); Throughput (Mbps); ECN marking

metrics. We aggregate measurements centrally for post-processing and analysis. Table 1 summarizes the key characteristics and statistical properties of the resulting dataset.

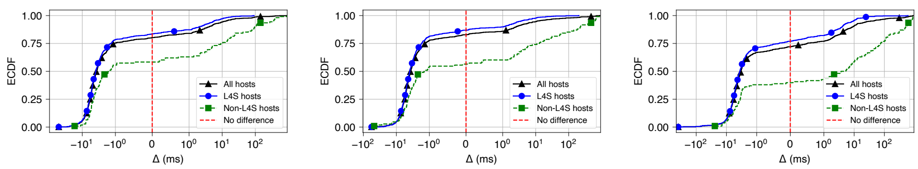
3.5 Limitations

Our measurement methodology has several limitations that constrain the generality of our findings. First, our deployment covers only a single ISP (Comcast) and uses L4S-capable home routers provisioned by the operator; thus, our observations reflect Comcast's access-network configuration and may not generalize to other ISPs. Second, we measure only Apple services, whose L4S and congestion-control behaviors are specific to Apple's ecosystem and may differ from other applications. Third, we cannot directly verify dual-queue AQM across the entire path; instead, we infer queue behavior from ECN markings and end-host congestion-control signals. Although end devices connect via Ethernet to the home router, bottlenecks may still arise upstream in the access or aggregation network, and our methodology cannot always isolate their location. Fourth, volunteer access-link speeds exceed those of many broadband users, reducing the likelihood of persistent queueing and limiting opportunities for L4S mechanisms to engage. Finally, we do not introduce controlled cross-traffic or engineered coexistence scenarios; results therefore reflect real-world but uncontrolled residential traffic mixes. These constraints do not invalidate our findings but do bound their applicability, and we make these assumptions explicit to guide interpretation.

Ethical Considerations. This work does not raise ethical concerns. Volunteers provided informed consent, and we collect only anonymized network measurements via triggered tests, not user application data or any personally identifiable information. All data is stored securely, and volunteers can withdraw from the study at any time.

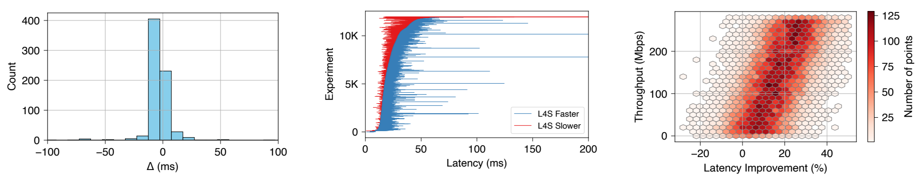
4 Results

We present empirical results from our deployment, analyzing over 60,000 paired measurements across 83 residential sites to assess L4S performance in production. We focus on tail latency (99th percentile) as the primary metric, given its



(a) Morning (b) Afternoon (c) Evening

Fig. 2: ECDFs of differences in 99th-percentile latency for Apple CDN download by time of day. Negative values indicate L4S reduces tail latency compared to classic congestion control. (50000 datapoints)



(a) Histogram of p99 latency differences (b) Per-experiment latency comparison (c) Correlation between latency improvement and throughput

Fig. 3: Apple CDN download per-experiment analysis showing tight distribution of improvements and consistent L4S advantage (blue lines indicate L4S reduces latency).

importance for interactive applications and user-perceived quality. We examine how L4S benefits vary across service types, time of day, and network conditions. We introduce the shorthand notation $\Delta = p_{99}(\text{L4S}) - p_{99}(\text{Classic Queue})$ as the difference in 99th-percentile latencies between L4S and classic queue.

4.1 Apple CDN Download: Bulk Transfer Performance

We begin with Apple CDN download tests to evaluate how L4S performs under sustained, throughput-oriented traffic, where long-lived flows are most likely to experience queue buildup and benefit from dual-queue scheduling. Grouping the data by time of day allows us to capture diurnal variations in access-network congestion, revealing whether L4S advantages persist across both lightly and heavily loaded periods.

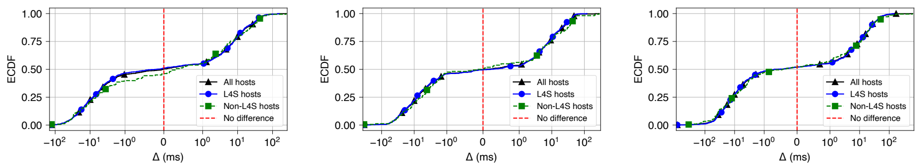
Aggregate Latency Improvements Across Time Periods. Figure 2 shows the empirical cumulative distribution of p99 latency differences for Apple CDN downloads across three time periods. Negative values indicate L4S reduces tail latency. L4S-enabled hosts (blue) consistently achieve lower p99 latency, with distributions shifted left of zero across all periods. During morning and afternoon, approximately 80% of L4S hosts experience latency reductions. Non-L4S hosts (green) show flatter distributions centered near zero, indicating negligible improvement. The aggregate across all hosts (black) follows the L4S trend, confirming benefits stem primarily from L4S-capable endpoints. Importantly, these improvements are not confined to a few favorable sites: similar left-shifted distributions are observed across the majority of deployment locations, suggesting that the latency improvements are robust and broadly consistent rather than driven by outliers or site-specific network conditions. During evening peak hours (06:00pm - 00:00am), absolute L4S improvements decrease slightly but remain substantial. Critically, the separation between L4S and non-L4S hosts widens considerably: the gap between blue and green curves increases on average from 8 ms (afternoon) to 20 ms (evening). This suggests that while absolute gains diminish under heavy load, L4S maintains a significant advantage over classic congestion control precisely when networks are most congested.

Consistency and Throughput Correlation. To validate the aggregate trends, we examine per-experiment variability through histograms of p99 latency differences and comparison plots summarizing the direction and magnitude of change across all experiments. These views reveal consistency and outlier behavior across our measurement campaign. The histogram (Figure 3a) shows a tight distribution centered just below zero, with most samples clustered between -10 ms and 0 ms. The pairwise plot (Figure 3b) confirms this with blue lines (L4S reduces latency) dominating across all sites and time periods. This consistency suggests that bulk download traffic reliably triggers queue buildup where L4S excels. Figure 3c shows that these latency improvements are not achieved at the expense of throughput. The scatter distribution shows no negative correlation between latency reduction and achieved throughput: experiments with strong latency gains sustain similar or even slightly higher throughput levels. This demonstrates that L4S achieves lower delay without compromising bulk-transfer efficiency, underscoring its effectiveness in balancing throughput and responsiveness under real-world conditions.

4.2 iCloud Download: Bursty Transfer Performance

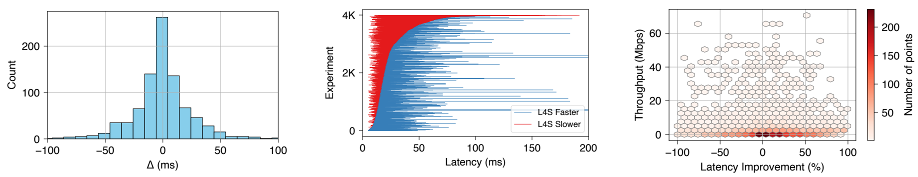
We next analyze iCloud download to assess how L4S behaves for short flows that differ fundamentally from sustained bulk transfers.

Aggregate Latency Improvements Across Time Periods. In contrast to Apple CDN downloads, iCloud traffic shows minimal L4S benefit (Figure 4). The L4S (blue) and non-L4S (green) distributions overlap substantially across all time periods, with both centered near zero difference. During afternoon and evening, the curves nearly coincide, with median change near 1 ms. This negligible improvement suggests that L4S provides little advantage for services that are tuned for synchronization. We attribute this to iCloud's traffic characteristics. Unlike sustained bulk transfers, iCloud is synchronization-oriented that consists of bursty, chunked uploads/downloads with application-layer rate limiting. These short bursts may not build sufficient queue depth for L4S's precise congestion signaling to yield measurable benefits. Additionally, iCloud traffic may use back-



(a) Morning (b) Afternoon (c) Evening

Fig. 4: ECDFs of differences in 99th-percentile latency for iCloud download by time of day. L4S and non-L4S distributions largely overlap. (35000 datapoints)

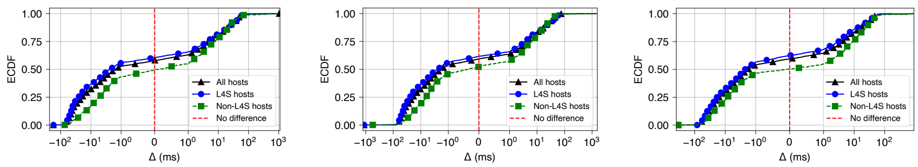


(a) Histogram of p99 latency differences (b) Per-experiment latency comparison (c) Correlation between latency improvement and throughput

Fig. 5: iCloud download per-experiment analysis showing symmetric distribution and mixed results (balanced blue/red lines indicate no systematic L4S benefit).

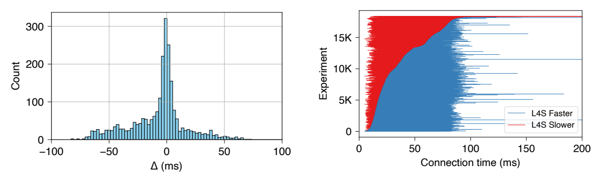
ground priority classes that avoid saturating links, further limiting opportunities for queue buildup. The lack of time-of-day variation (similar performance during peak and off-peak hours) supports this interpretation; iCloud traffic does not stress the network enough to benefit from improved queue management.

Consistency and Throughput Correlation. The histogram (Figure 5a) exhibits a wider, symmetric distribution centered near 0 ms for iCloud. Both improvements (negative) and degradations (positive) appear with roughly equal probability. The pairwise plot (Figure 5b) shows balanced blue and red lines, confirming no systematic L4S advantage. The high variability suggests that iCloud performance depends on factors orthogonal to queue management, *i.e.*, application-layer behavior and traffic patterns. Figure 5c reveals no consistent correlation between throughput and latency improvement, indicating that L4S effects are largely independent of bulk-transfer rate. Unlike Apple CDN downloads, where latency gains align with sustained throughput, iCloud experiments show a highly scattered pattern: several tests even display positive latency differences (i.e., higher latency under L4S) at moderate throughputs. This dispersion suggests that the short-lived, bursty nature of iCloud traffic prevents L4S from establishing stable dual-queue dynamics, limiting its ability to realize queueing-delay reductions. Overall, the absence of a clear trend reinforces that L4S benefits are workload dependent—effective for steady, congestion-prone flows but less impactful for sporadic, background exchanges.



(a) Morning (b) Afternoon (c) Evening

Fig. 6: ECDFs of differences in 99th-percentile call establishment time for FaceTime by time of day. L4S consistently reduces latency, especially during peak hours. (35000 datapoints)



(a) Histogram of p99 call establishment time differences

(b) Per-experiment call establishment time comparison

Fig. 7: FaceTime per-experiment analysis showing broader distribution with heavy left tail and strong L4S advantage during peak periods

4.3 FaceTime: Real-Time Interactive Performance

Having examined L4S behavior for throughput-oriented services, we now turn to FaceTime, a latency-critical application. FaceTime's performance is dominated by real-time delay sensitivity, making it an ideal test case to assess L4S benefits under human-perceptible latency constraints.

Aggregate Latency Improvements Across Time Periods. Figure 6 shows consistent L4S benefits across all time periods. The effect is relatively stable across morning, afternoon, and evening, with improvements of roughly 8 ms compared to classic, suggesting the service benefits from L4S even under moderate load. Non-L4S hosts (green) show substantially worse performance, particularly during afternoon and evening periods where distributions shift rightward (indicating latency increases). The separation between L4S and non-L4S curves widens during peak hours, reaching 10 ms in the evening. This pattern suggests that interactive real-time applications derive greater relative benefit from L4S under congestion—precisely when low latency is most critical for user experience.

Consistency and Per-experiment Validation. Figure 7a displays a broader distribution with heavy left tail extending to −50 ms, indicating substantial improvements for a subset of experiments. Figure 7b shows a clear dominance of blue lines, indicating that L4S consistently improves connection setup latency across most experiments. Improvements are concentrated around the lower latency range (below 100 ms), consistent with FaceTime's real-time nature and short flow durations. The limited number of red outliers suggests that L4S rarely degrades performance, and when it does, the magnitude of regression is small. Overall, the results confirm that L4S provides reliable latency reduction for interactive, delay-sensitive traffic, validating its suitability for real-time applications like video conferencing.

Main takeaways: *Across all analyses, our findings reveal that L4S effectiveness strongly depends on the traffic characteristics of each service. For Apple CDN downloads, L4S consistently lowers tail latency for bulk transfers, confirming that dual-queue mechanisms are most effective when flows sustain congestion long enough for ECN signaling to stabilize. For iCloud downloads, however, the impact remains negligible, as short, bursty synchronization flows seldom trigger persistent queue buildup. Finally, FaceTime demonstrates meaningful latency reductions, particularly during peak hours, validating L4S's value for interactive, real-time communication. Overall, these results show that L4S delivers tangible benefits where persistent congestion or latency sensitivity dominate performance, but offers limited gains for transient, low-volume traffic.*

5 Conclusion

We present the first large-scale, in situ measurement study of L4S in production residential broadband networks. Deploying 83 devices across US Comcast networks and conducting over 60,000 paired experiments, we quantify L4S performance for three representative Apple services: Apple CDN, iCloud and FaceTime.

Our key findings reveal substantial but service-dependent L4S benefits. FaceTime and CDN downloads experience consistent tail latency reductions up to 25%, with advantages most pronounced during evening peak hours (20 ms for CDN, 10 ms for FaceTime) when non-L4S performance degrades. In stark contrast, iCloud shows negligible benefit, with L4S and non-L4S distributions overlapping across all periods. This disparity reveals a critical insight: L4S effectiveness requires sustained queue buildup; while bursty, application-paced traffic cannot benefit from improved queue management. The widening performance gap during congestion suggests L4S successfully isolates latency-sensitive flows from the effects that degrade classic congestion control.

Our results provide evidence of L4S's practical value for interactive and bulk transfer workloads, while identifying deployment contexts where benefits may not materialize. As L4S adoption expands in commercial services and ISP infrastructure, our measurement methodology and empirical findings offer guidance for operators and application developers assessing L4S deployment value.

References

1. Apple Inc.: Testing and Debugging L4S in Your App. https://developer.apple.com/documentation/network/testing-and-debugging-l4s-in-your-app
2. balena: OpenBalena. https://open-balena-docs.balena.io/
3. BoruOljira, D., Grinnemo, K.J., Brunstrom, A., Taheri, J.: Validating the sharing behavior and latency characteristics of the l4s architecture. SIGCOMM Comput. Commun. Rev. pp. 37–44 (2020)
4. Briscoe, B., Schepper, K.D., Bagnulo, M., White, G.: Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture. RFC 9330 (Jan 2023). https://doi.org/10.17487/RFC9330, https://www.rfc-editor.org/info/rfc9330
5. Comcast: What is Latency. https://corporate.comcast.com/stories/latency-speed-explainer
6. Gettys, J., Nichols, K.: Bufferbloat: Dark buffers in the internet. Communications of the ACM **55**(1), 57–65 (2012)
7. Graff, P., Marchal, X., Cholez, T., Mathieu, B., Tuffin, S., Festor, O.: Improving Cloud Gaming traffic QoS: a comparison between class-based queuing policy and L4S. In: 2024 8th Network Traffic Measurement and Analysis Conference (TMA). pp. 1–10. IEEE (2024)
8. Monteiro, L.V., Simão, V.S., Lira, R.d.B., de Almeida, L.C., Gomes, R.D., Ditarso Maciel, P.: L4s in private 5g industrial networks: A case study for real-time video transmission in programmable networks. In: 2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN). pp. 1–4 (2024). https://doi.org/10.1109/NFV-SDN61811.2024.10807467
9. Nádas, S., Gombos, G., Fejes, F., Laki, S.: A congestion control independent l4s scheduler. In: Proceedings of the 2020 Applied Networking Research Workshop. pp. 45–51 (2020). https://doi.org/10.1145/3404868.3406669
10. Nguyen, H.N., Mathieu, B., Letourneau, M., Doyen, G., Tuffin, S., Oca, E.M.d.: A comprehensive p4-based monitoring framework for l4s leveraging in-band network telemetry. In: NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium. pp. 1–6 (2023). https://doi.org/10.1109/NOMS56928.2023.10154331
11. Nvidia: What is the L4S setting in the GeForce NOW streaming quality menu. https://nvidia.custhelp.com/app/answers/detail/a_id/5522
12. Schepper, K.D., Briscoe, B.: The Explicit Congestion Notification (ECN) Protocol for Low Latency, Low Loss, and Scalable Throughput (L4S). RFC 9331 (Jan 2023). https://doi.org/10.17487/RFC9331, https://www.rfc-editor.org/info/rfc9331
13. Schepper, K.D., Briscoe, B., White, G.: Dual-Queue Coupled Active Queue Management (AQM) for Low Latency, Low Loss, and Scalable Throughput (L4S). RFC 9332 (Jan 2023). https://doi.org/10.17487/RFC9332, https://www.rfc-editor.org/info/rfc9332
14. Srivastava, A., Fund, F., Panwar, S.S.: An experimental evaluation of low latency congestion control for mmwave links. In: IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 352–357 (2020). https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162881

A Controlled vs. Inferred Assumptions

This appendix summarizes which aspects of our ECN/L4S measurement methodology are *controlled* (i.e., known with certainty based on our deployment and instrumentation) and which aspects are *inferred* from observable traffic characteristics. These apply uniformly across all three workloads (Apple CDN, iCloud, and FaceTime).

A.1 Controlled Aspects

- **L4S-capable home routers.** Comcast provides an up to date list of volunteers provisioned with home routers that are L4S-capable.
- **L4S-capable measurement device.** All experiments run on Raspberry Pi devices whose Linux kernel includes L4S-capable congestion control and ECN support.
- **Explicit enable/disable control for L4S.** Our measurement client controls whether L4S is enabled or disabled at the end device. This configuration is set deterministically by our code for each experiment.
- **Apple services implement L4S.** The Apple CDN, iCloud, and FaceTime services we measure implement L4S semantics and set ECT(1) for L4S-capable flows.

A.2 Inferred Aspects

- **End-to-end L4S support.** We do not know whether all intermediate hops between Apple servers and our Raspberry Pi devices support or preserve L4S behavior.
- **Queue behavior and AQM configuration.** We infer queueing behavior—including possible dual-queue operation—indirectly from ECN codepoints observed in received packets and from endpoint TCP state.
- **Bottleneck.** Congestion may arise upstream in the access or aggregation network, and we cannot always isolate the precise bottleneck.

B Summary Statistics and Baseline Distributions

To complement our tail-latency analysis and provide baseline context, we report descriptive statistics for each service under both L4S and Classic configurations. These values summarize typical latency levels and their variability. In Table 2, we present the mean, median, p90, interquartile range (IQR), and standard deviation (StdDev) for Apple CDN downloads, iCloud, and FaceTime call establishment time. These baseline statistics contextualize the tail behavior discussed earlier in Section 4 and help quantify how representative the tail differences are relative to the overall distribution.

Table 2: Baseline summary statistics (ms).

Service	Configuration	Mean	Median	p90	IQR	StdDev
Apple CDN	Classic Queue	25.47	22.66	38.40	9.52	10.40
	L4S	24.65	18.97	35.92	10.20	39.99
iCloud	Classic Queue	27.46	18.34	47.98	14.92	58.89
	L4S	24.47	18.64	45.57	13.83	34.68
FaceTime	Classic Queue	39.28	31.98	76.87	42.36	31.57
	L4S	33.66	24.3	71.86	28.39	28.84