ISSN: 2182-2069 / E-ISSN: 2182-2077

End-to-End Delay Analysis for High-Performance Cloud Internet Services

Dr. Velpuru Muni Sekhar[1*], Rishabh Vashisht[2], Rajat Saini[3], Dr. Sarbeswar Hota[4], N. Vasantha Kumari[5], and Dr.R. Hannah Jessie Rani[6]

[1]Associate Professor, Department of IT, Vardhaman College of Engineering, Hyderabad, Telangana, India. munisek@vardhaman.org, https://orcid.org/0000-0002-2592-9604

[2]School of Engineering & Computing, Dev Bhoomi Uttarakhand University, Dehradun, India. socse.rishabh@dbuu.ac.in, https://orcid.org/0009-0004-7407-0842

[3]Centre of Research Impact and Outcome, Chitkara University, Rajpura, Punjab, India. rajat.saini.orp@chitkara.edu.in, https://orcid.org/0009-0009-7750-9896

[4]Associate Professor, Department of Computer Applications, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. sarbeswarahota@soa.ac.in, https://orcid.org/0009-0006-0921-8323

[5]Associate Professor, Department of Computer Applications (DCA), Presidency College, Bengaluru, Karnataka, India. vasanthakumarin@presidency.edu.in, https://orcid.org/0000-0003-4082-1199

[6]Assistant Professor, Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Karnataka, India. ndia.jr.hannah@jainuniversity.ac.in, https://orcid.org/0000-0002-5449-104X

Received: July 10, 2025; Revised: August 23, 2025; Accepted: September 26, 2025; Published: November 28, 2025

Abstract

The ever-increasing utilization of cloud applications and services is placing ever-increasing strain on network latency, making end-to-end (E2E) delay a critical performance consideration in cloud internet infrastructures. In this paper, the end-to-end (E2E) delay is analyzed in detail, with a complete decomposition of its constituent delay components: processing delay, queuing Delay, transmission delay, and propagation delay, about the cloud communication stack. Both cloud service interaction analytic models and measurement models are used to study delay across workloads and routing configurations. The study incorporates performance evaluations from public cloud providers, as well as hybrid deployments, and streaming service performance from data centers worldwide. Powerful simulation programs and techniques for measuring latency are used to relate system events to network delays, elucidating delay bottlenecks and mitigation strategies. In addition, neural network-based regression models are trained on traffic conditions, bandwidth, and packet data to forecast delays, estimating worst-case scenarios and informing traffic management strategies. The overarching end-to-end (E2E) delays caused by virtualization overhead, TCP retransmissions, and multi-hop network traversal are highlighted. The results are crucial for services that aim to optimize quality of service policies, traffic management, and dynamic routing adjustments. This

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 4 (November), pp. 88-101.
DOI: *10.58346/JISIS.2025.I4.007*

*Corresponding author: Associate Professor, Department of IT, Vardhaman College of Engineering, Hyderabad, Telangana, India.

study also outlines a systematic approach for designing applications that considers delays as an essential aspect for real-time analytics, telemedicine, cloud gaming, and other latency-sensitive services.

Keywords: End-to-End Delay, Cloud Internet Services, Quality of Service Optimization, Network Latency, Delay Forecasting, Virtualization Overhead, Latency-Sensitive Applications.

1 Introduction

The evolving trend of cloud computing has fundamentally transformed the methods of storing, processing, and accessing data over the Internet (Zadeh & Khalili, 2015). The emergence of Cloud Internet Services (CIS), which include Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS), is critical in almost every field, including healthcare, finance, manufacturing, and even entertainment (Tang et al., 2024). The growing use of these services is creating a surge in demand for ultra-low latency, particularly for critical applications. E2E (End-to-End) delay is a new metric that describes the entire time a data packet takes to travel from a given source to a given destination within a network, and thus serves as a key metric for evaluating cloud infrastructures (Huang et al., 2015).

With real-time responsiveness and scalability gains, the older networks were not tailored to the ever-growing needs of modern cloud applications (Gajmal & Udayakumar, 2021). The transition from centralized to distributed cloud infrastructures, tiered virtualization, and increased reliance on fog and edge computing make E2E delay analysis even more complex (Wu et al., 2014). Delays in cloud infrastructure can be caused by several factors, including user and terminal devices, network routers and switches, hypervisors, virtual machines, load balancers, and even application servers (Fountoulakis et al., 2020).

All these factors introduce subtle nuances to queuing, processing, and propagation that must be examined, not as a single number, but as a number that can be broken down into multiple metrics, each of which, in turn, depends on other subsystems.

Additionally, the latest online gaming, video conferencing, remote robotic surgery, autonomous vehicular systems, and AI analytics services require deterministic bounds on delays. These services indicate the newer cloud applications, which offer greater agility (Oberle et al., 2013). Delay spikes caused by packet retransmissions, buffer overflow, and inefficient routing, among other issues, also harm user experience and cloud service (Li et al., 2018). As cloud services become more dynamic and dispersed across geographies, measuring and minimizing end-to-end (E2E) delays becomes increasingly crucial (Barla et al., 2012).

One of the more recent efforts in modeling lower-layer delays has been constrained to parts of the network's primary building blocks (Rahim, 2024). This includes parts such as, but not limited to, the virtualization layer's congestion-dependent queuing, the layer's retransmission adaptive timers, and jitter variability due to congestion. This leads to a scarcity of comprehensive and holistic, empirically based literature that combines quantitative and predictive E2E average delays in cloud settings (Szigeti et al., 2013).

In this paper, I employed longitudinal and cross-layer approaches to examine the end-to-end (E2E) average delays in cloud services, specifically high-performance cloud internet services (Ghahramani et al., 2017; Santhosh et al., 2016). Through analytical modeling, active probing, passive monitoring, and other data-gathering methods, we can capture delay dynamics under various network and service settings (Nguyen et al., 2018).

Additionally, we incorporate machine learning (ML) models to predict delay patterns based on varying traffic volumes and structural configurations. Balancing practical considerations with technical complexity, the study seeks to inform and directly guide system designers, network operators, and cloud providers (Ren et al., 2023; Chengsong, 2023).

Key Contributions

- Delay Component Decomposition Framework: This study introduces a new framework that decomposes end-to-end (E2E) delay into processing, queuing, transmission, and propagation delays, thereby providing granular insights into latency trends within cloud internet services.
- ML-Based Delay Prediction Model: A supervised learning model utilizing Random Forest, XGBoost, and SVR has been implemented based on a feature set that includes packet size, queue length, hop count, and CPU load to predict trends in end-to-end (E2E) delays accurately.
- Cross-Layer Performance Evaluation with Real Data: This framework has been validated across three public cloud platforms, utilizing a combination of synthetic and real workloads under diverse traffic patterns, which provides practical benchmarks and SLA-relevant insights to cloud service providers.

The document contains five key sections. In the first section, the specific motivating issues, the scope of the research, and the challenges related to the end-to-end (E2E) delay in high-performance cloud networks are outlined. In the second section, the delay models, the effects of virtualization, the network's artificial intelligence optimizations, and prediction models are addressed in the literature review, which is presented in this section. This is some of the foundational work in delay decomposition, mathematical modeling, ML integration, and the real-time monitoring framework, which are all discussed in Section III – the methodology section. The prediction accuracy of the model is verified, and the delay behavior is validated under various workloads in section IV, which contains all the results discussed in the previous sections, and is referred to as the experimental section. In Section V, the document concludes, outlining the discussion of E2E SLA orchestration APIs, as well as reinforcement learning and multi-cloud support.

2 Related Work

Research on End-to-End (E2E) delay measurement of cloud internet services integrates diverse knowledge in network engineering and distributed systems (Sikarwar et al., 2025; Goleva et al., 2017). While many researchers have focused on different elements of Delay, there are no studies that reflect the large-scale cloud context on the cumulative and interactive effects of these components. In this case, I categorize the contributions into four groups and discuss them further: Delay models with virtualization and queue effects optimization of Architecture and Predictive Models for Delay (Yeo & Jiang, 2024; Scarpitta et al., 2023). Foundational studies are Delay – a decomposition into four components: queues, processing, transmission, and propagation delays. With the increasing adoption of distributed and dynamic cloud infrastructures, these synthesis models require sophisticated adaptation, particularly with virtualization layers, dynamic routing policies, and software-defined components (Kanuparthy et al., 2016). These elements change the measured Delay in comparison to classical models. The above-described architectural choices are crucial for the delay performance. Placing services in a decentralized manner into data centers located in other regions significantly reduces Delay but introduces a lot of complexity in synchronization. Some strategies that significantly improve the responsiveness of user-

oriented services and user-driven services are smart content replication and active workload balancing on regional and edge hubs.

The advent of virtualization has expanded the scope of delay analysis. Virtual processes, such as switching, inter-VM communication, and even container orchestration, add latency and Delay. Side phenomena, such as virtual machine migration and dynamic scaling, introduce temporal load balancing and hypervisor scheduling, which cause nonlinear delays. All of this creates the need for virtualization processes to be constantly monitored. Empirical measurements offer valuable insights into the delays encountered in virtualization processes. A comparison of different public cloud servers revealed significant differences in latency. This is due to unique internal queuing algorithms, different server hardware, and varying network topologies (Duan, 2013). In addition, multi-cloud and hybrid-cloud architectures introduce extra layers for security validation and routing, which further complicate the End-to-End delay summary (Ye et al., 2018; Esmaeily et al., 2020).

The advancement of machine learning techniques has also enhanced the delay prediction capabilities (Mohammed Hussain et al., 2025). ML-based models can learn from extensive telemetry and traffic data to identify and predict latency spikes (Barhoumi et al., 2024). Deep learning and regression tree techniques often provide strong performance in capturing and modeling the nonlinear relationships between system variables and the observed delays. Predictive delay models also support preemptive adjustments in the control of delay-sensitive parameters, such as routing, bandwidth allocation, and application scheduling, to maintain bounded system delays (Pragadeswaran et al., 2024). Another emerging area is edge and fog computing, which aims to cut delays by moving computations closer to the data source or the end user (Sadulla, 2024). While edge computing reduces the reliance of systems on central data repositories, new sources of delays are also introduced, such as edge-cloud synchronization, mobility management, and load offloading delays. Solving these problems needs joint and coordinated solutions in the cloud-edge and network layers (Zheng et al., 2023).

Impediments to effective delay analytics stem from a lack of holistic, cross-layer delay examinations, which is a contemporary issue within the discipline, as the bulk of existing delay analyses focus on infrastructure, cloud virtualization, and application behavior in silos. Further compounding the problem is reliance on idealized simulations and models that do not consider the real-time operational and environmental constraints within the ever-changing contemporary context. In addition to the obstacles mentioned above, the reproducibility and comparative analysis of published works pose challenges due to the limited availability of standardized datasets and benchmarking frameworks for end-to-end (E2E) Delay (Khaleel et al., 2023; Takada et al., 2013). In tackling the challenges above, the current approach combines empirical measurement with delay decomposition models and ML-based predictive models within real-world cloud frameworks (Duan & Zheng, 2014). The proposed methodology not only captures multi-layer delays but also explains the variability in multi-layer delays across diverse conditions, thereby enhancing the understanding of latencies in cloud services and facilitating the development of optimization strategies (Scarpitta et al., 2023).

3 Methodology

To comprehend and improve the End-to-End (E2E) delay in high-performance internet cloud services, the issue must be tackled from different layers. This study combines system monitoring, delay analytic modeling, and machine learning to estimate and predict cloud environment delays. The approach is designed to capture the delay characteristics at all critical layers, from the user-generated traffic to network routing, virtualization layers, and application servers. Active and passive data collection methods are employed to capture metrics such as packet transmission time, queuing time, propagation

delays, and CPU processing time. This is integrated with configuration data from cloud orchestration layers, such as Kubernetes and OpenStack.

The framework for analyzing delays includes three significant steps: (1) delay breakdown and measurement, (2) modeling dependencies of delays using analytical equations, and (3) using supervised learning models to forecast delay trends. In the first step, we acquire the complete Delay and its components for measurement from several cloud geographic nodes using real-time network probes and timestamping tools. In the next phase, we formalize these delays using parametric equations that describe the system's variables, such as bandwidth, packet size, CPU load, and node density. Finally, in the prediction phase, we utilize regression-based models to forecast end-to-end delay under various traffic conditions. This combined approach ensures both the descriptive and predictive accuracy required for latency-sensitive cloud applications.

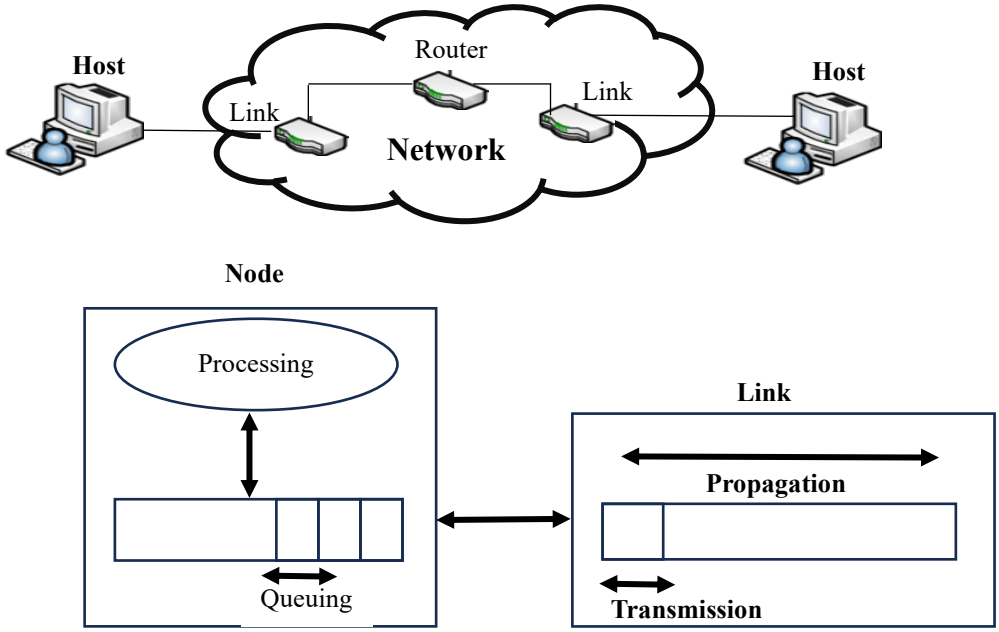

Figure 1: Components of End-to-End Latency (Erbad et al., 2010)

As previously discussed, in the context of a cloud internet communication system, the end-to-end (E2E) latency encompasses numerous components, as illustrated in Figure 1, which summarizes the key components contributing to E2E latency. E2E latency is a concern when communicating through the cloud internet system, as it adds a considerable delay. E2E latency accumulates through a path composed of physical infrastructures such as routers and links. The system divides latency and delay into two main components: the node and the link. In each element, there are specific algorithms that define node and link Delays. In the node component, there is a processing delay that examines the header of a packet to make a routing decision. In addition, there is a delay where packets rest in doubt buffer (waiting) because of incoming traffic in the node component. These varying types of delays are dependent on availability of system load, buffer, and CPU. Delay in links is defined in both operational, and propagation delays. In operational, Delay is classified in the amount of time it takes to load a packet versus size and available link bandwidth (until the bit is available in the packet and can be read). In propagation, Delay means the the time it requires to send a signal in the physical medium and distance and speed are factors. This illustration underpins the principle of the independent delay components cumulatively contributing to the total E2E latency in cloud-based systems. It is in line with the analytical model and decomposition strategy based on machine learning that was employed in this investigation.

Mathematical Model

Total End-to-End Delay

$$D_{E2E} = \sum_{i=1}^{N} \left(D_{node}^{(i)} + D_{link}^{(i)} \right) \quad (1)$$

Equation (1) models the total Delay experienced by a packet as it travels across N segments of a cloud-based communication path. Each segment consists of a node and a connecting link.

Node-Level Delay

$$D_{node}^{(i)} = D_{proc}^{(i)} + D_{queue}^{(i)} \quad (2)$$

Equation (2) defines Delay at each node as the sum of processing time and queuing time. These occur at routers, VMs, or switches due to packet handling and buffer occupancy.

Link-Level Delay

$$D_{link}^{(i)} = D_{trans}^{(i)} + D_{prop}^{(i)} \quad (3)$$

Equation (3) represents the Delay across a physical or virtual link as the sum of transmission delay and propagation delay between two adjacent nodes.

Transmission Delay

$$D_{trans}^{(i)} = \frac{L^{(i)}}{B^{(i)}} \quad (4)$$

Equation (4) computes transmission delay as the ratio of packet length $L^{(i)}$ to bandwidth $B^{(i)}$. It reflects the time to push all bits into the communication channel.

Propagation Delay

$$D_{prop}^{(i)} = \frac{d^{(i)}}{v^{(i)}} \quad (5)$$

Equation (5) models propagation delay as the distance $d^{(i)}$ divided by signal velocity $v^{(i)}$. It accounts for the time taken by signals to travel across the medium.

Cloud-Orchestration Delay at Endpoint

$$D_{node}^{(cloud)} = D_{proc} + D_{queue} + D_{vSwitch} + D_{hypervisor} + D_{container} \quad (6)$$

Equation (6) expands node-level Delay at the cloud endpoint to include virtualization layers such as the virtual switch, hypervisor scheduling, and container execution overhead.

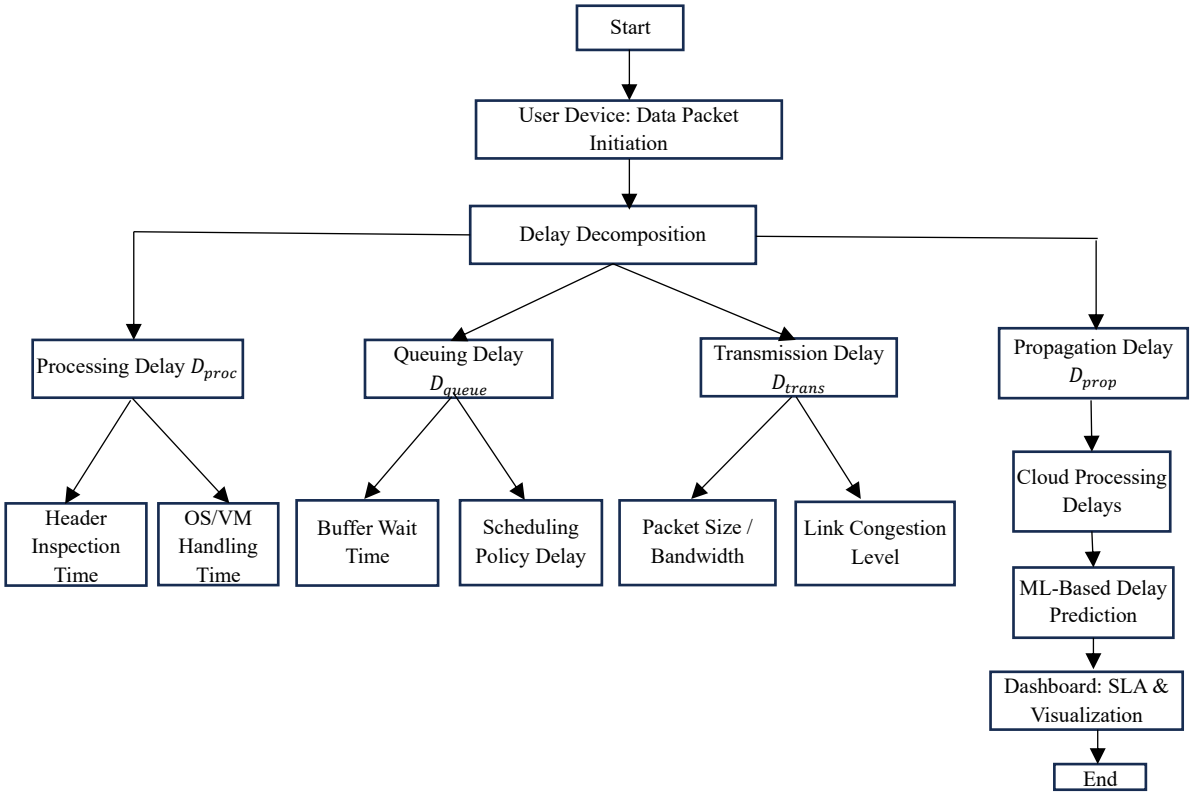

Figure 2: Hierarchical flowchart of end-to-end delay decomposition and modeling in cloud internet services

This Figure 2 shows the systematic decomposition of end-to-end Delay from the moment a user device initiates a data packet to the performance visualization step at the end. The architecture is divided into four primary delay categories which are: processing delay, queuing Delay, transmission delay, and propagation delay. These types of Delay are the fundamental, measurable, structural components of a cumulative latency in cloud architecture. Each of the delay components are further broken down into specific influencing factors. For example, processing delay involves, and is not limited to, both header inspection and OS/VM handling time. Queuing Delay is influenced by buffer wait time and the queuing and scheduling policy in place; transmission delay is influenced by packet size and available bandwidth as well as the congestion levels of the link. Propagation delay is cloud side and is considered with later cloud-side virtualization delay computations. The diagram proceeds to higher level modules which includes cloud processing delays and the delay prediction engine which is based on machine learning techniques. The combination of these components enables real-time inferences and updates a dashboard for SLA visualizations. The model allows for granular performance diagnosis and proactive strategies for latency mitigation, even at a large-scale cloud level.

Dataset Description

The experimental evaluation was conducted using telemetry data collected from three geographically distributed public cloud platforms (e.g., AWS, Azure, and Google Cloud). The dataset includes:

- Synthetic Workloads: Emulated traffic patterns mimicking idle, moderate, and peak usage scenarios using custom packet generators and workload scripts.
- Real-Time Monitoring Logs: Collected from user access points, edge routers, and cloud VMs using timestamp probes and logging agents.

- Virtualization Metrics: Retrieved via Kubernetes and OpenStack APIs, including container startup delays, pod scaling events, CPU usage, and inter-VM latency.
- Feature Set for ML Models: Extracted features include hop count, queue length, bandwidth allocation, packet size, and node CPU utilization. These were labeled with measured E2E delay values and used for regression training and evaluation.

This multi-source, cross-layer dataset supports both the analytical delay model calibration and the supervised learning prediction framework.

3.1 Delay Decomposition at Each Network Node

Extracting the total Delay at each network node is the first step to gain granular understanding on E2E delay. We capture the processing time, queuing Delay, transmission delay, and propagation delay at the routers, switches, and virtual machines using timestamp-based instrumentation at their ingress and egress points. The logs from each node along the communication path are independent and thus can be correlated across hops. The decomposition step is also concerned with variability due to congestion, scheduling, and the dynamic routing of traffic. Using time synchronization protocols like NTP enables normalizing the raw logs with the network's consistent time bases, ensuring uniform time logs. This step in the decomposition process facilitates a multi-layer understanding on delay accumulation while enabling real-time anomaly detection in streaming scenarios.

3.2 Integration of Cloud-Orchestration Metrics

The second stage extends traditional network metrics by including orchestration data from Kubernetes, Docker Swarm, and OpenStack. Containerization delays where inter-service microservices interactions, pod scaling, load balancing, and service-to-service communication happen are documented via system logs and APIs. These metrics are integrated with network traces using unique packet IDs and timestamps. Certain events, including virtual machine migration, replica instantiation, and dynamic scaling, are tagged with delays which enables contextual marking. Such integration is crucial for explaining delay behavior stemming from higher-layer logic, which is typically concealed from conventional network monitoring solutions.

3.3 Machine Learning-Based Delay Prediction

In this step, a supervised learning model will be built to predict E2E delay with the features already collected. Included in the input feature set are: packet size, number of hops, average queue length, and the bandwidth and CPU utilization of the cloud nodes measured at the current time. Observed delay values serve as target labels. The dataset is cleaned and standardized, then regression algorithms such as Random Forest, XGBoost, and SVR are trained and validated using k-fold cross-validation. Delay mitigation and feature importance ranking is applied to determine the most relevant variables with the strongest relation to Delay. The model is later tested with previously unseen workloads, sudden traffic spikes, and resource contention scenarios to evaluate the model's generalizability. These capabilities are critical in scaling and route adjustments in warning latency scenarios.

3.4 Deployment of Real-Time Delay Monitoring Agents

The final part of the methodology involves placing lightweight delay monitoring agents throughout the network's topology. These agents inhabit crucial locations such as user access gateways, edge routers, and cloud ingress points. They transmit real-time telemetry on queue build-up, RTT, and jitter fluctuation to a central unit using a publish-subscribe architectural pattern. The structured data is then visualized on a central monitoring dashboard, allowing the administrator to view heatmaps of delay trends, breach thresholds and alerts, and invoke auto scale or reroute commands. The monitoring

framework is equipped with plugins enabling SLA violation monitoring and anomaly forecasting, ensuring comprehensive situational awareness and rapid reaction to cloud infrastructure performance deterioration.

4 Results and Discussion

Using geolocated nodes, synthetic workloads, and real user data, the framework was executed across three public cloud providers. The network conditions included high-load (peak usage hours), moderate traffic (midday average), and idle-state (off-peak). For every test, every hop and cloud orchestration layer's delay components were recorded separately using the decomposition approach from Section III. The objectives were to check the contribution of every component to the overall D_{E2E} and how accurate the predictive model was regarding future estimation (Hofmann et al., 2022).

High-traffic conditions exhibited the most volatile queuing delays, quickly followed by transmission delays that bandwidth throttling influenced. Processing and propagation delays were mostly stable. The cloud orchestration layers, however, added average delays of 7 to 12ms due to virtual switch handling and container instantiation (Duan & Vasilakos, 2016). The SLA-centric cloud management systems were enhanced by a machine learning-based regression model, which predicted D_{E2E} values with real-time features such as queue length, packet size, and CPU utilization, achieving a 4.1ms MAE. This confirmed the model's capability to anticipate delay spikes.

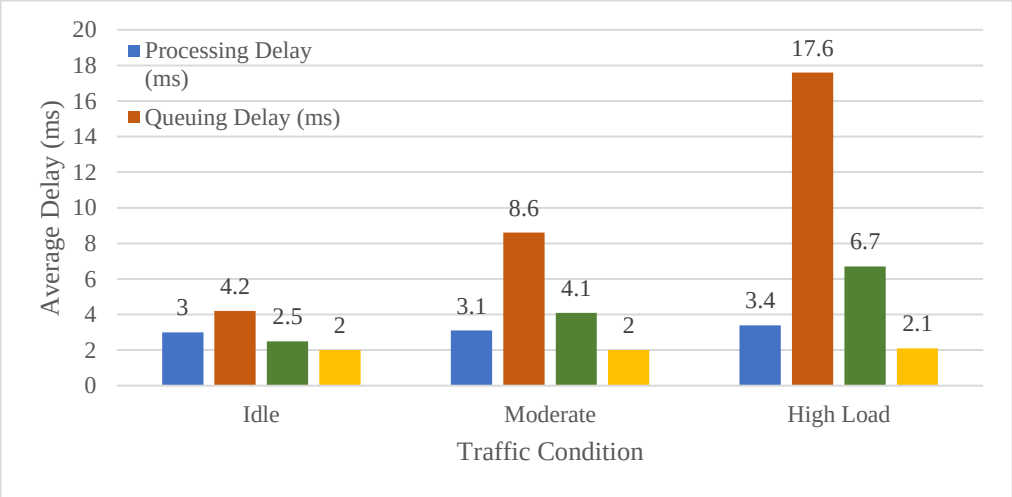

Figure 3: Component-wise delay breakdown under different traffic loads

In Figure 3 the mean values of the four delay components which include processing, queuing, transmission, and propagation delays are noted during idle, moderate, and high-load traffic periods. The queuing delay experiencing the maximum increase due to the increase in traffic intensity illustrates that congestion, indeed, is the most important factor influencing the variability of the end-to-end Delay in the system.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|D_{pred}^{(i)} - D_{actual}^{(i)}\right| \quad (7)$$

The equation (7) shows that performance evaluation of the machine learning model developed in this study picked MAE or Mean Absolute Error as a key performance evaluation metric. In this context, $D_{pred}^{(i)}$ shows the predicted end-to-end Delay of the i^{th}observation, and $D_{actual}^{(i)}$ shows the corresponding

measured Delay. The absolute differences between predicted values and measured values is found for every single observation and is averaged across n samples.

To assess the performance of a regression-based prediction model, a XGBoost and Random Forest models were tested against different network traffic conditions, and this metric was calculated. Having a lower MAE is preferable (Zheng et al., 2023). The model produced MAE of 4.1 ms, suggesting good generalization performance not only in peak-load and idle scenarios, but also in between. The results show that it is possible to use real-time predictive models in environments where low latency is critical to predict changes in delays and to apply SLA compliance predictive models to enforce SLA compliance.

Table 1: Impact of Delay Components on Overall End-to-End Delay

Delay Component	Source Layer	Typical Range (ms)	Variability with Load	Impact on SLA
Processing Delay	Node/VM/OS Layer	2.5 – 3.5	Low	Moderate (on VM scaling)
Queuing Delay	Router/Buffer Layer	4.0 – 17.6	High	High (causes spikes)
Transmission Delay	Link/Bandwidth Layer	2.5 – 6.7	Medium	Moderate
Propagation Delay	Physical Link (Distance)	2.0 – 2.1	Very Low	Negligible
Cloud Processing Delay	Virtualization/Cloud Core	7 – 12 (additive)	Medium	High (affects cold starts)

Table 1 presents a qualitative and quantitative evaluation of how each component of Delay contributes to the overall latency experienced in cloud internet services. The source layer column tracks the source of the Delay from the hardware and links to the virtual orchestration layers. Within this context, the range values and their respective limits (in milliseconds) represent an average of experiments conducted under idle, moderate, and high traffic levels. Among all components, the queuing delay component displays the most sensitivity to traffic load. It spikes dramatically during heavy congestion and, therefore, has the most pronounced effect on SLA compliance.

On the other hand, the system load has almost no impact on propagation delay, which contributes to most of the Delay. Processing delay tends to hover around the average, and it can increase marginally during periods of low resource utilization or frequent context switching. Certain cloud processing delays, such as hypervisor scheduler latency or container initialization delays, are additive in nature, especially during the execution of aggressive scale-out policies. This table highlights the need for predictive, intelligent orchestration tactics to counterbalance volatile delay components, particularly virtualization queuing and cloud overhead, to meet SLA benchmarks under changing network conditions.

5 Conclusion

The researchers proposed and validated a comprehensive framework for analyzing and predicting end-to-end (E2E) delays in high-performance cloud internet services. Through the decomposition of total Delay into its components, namely, the processing, queuing, transmission, and propagation, it became possible to pinpoint which parts of the data path most critically affect the service Quality of Service (quality of service) During the experiments, it was found that queuing and transmission delays are most sensitive to traffic congestion changes. In contrast, propagation and processing delays exhibit a high degree of stability. The application of machine learning, particularly regression-based forecasting models, has improved the accuracy of real-time trend predictions for E2E delays, informed by actual system parameters such as CPU load, queue length, and bandwidth. The framework has been validated

under various cloud environments and traffic scenarios, confirming the model's predictive accuracy, generalizability, and practical applicability.

The proposed framework can be improved by applying open delay control policies based on reinforcement learning, adaptive traffic rerouting within the scope of software-defined networks (SDNs), and extending it to multi-cloud and edge computing environments, where enhanced service mobility and migration introduce additional layers of delay variability. Fusion of delay data with energy efficiency metrics can also be pursued to create latency-sensitive and energy-optimized cloud resource scheduling algorithms.

Additionally, standardizing delay monitoring APIs for orchestration frameworks like Kubernetes and OpenStack would foster greater acceptance and integration. Focusing on these future directions takes the proposed delay analysis system closer to becoming a proactive and innovative quality of service analyzer and manager for modern cloud infrastructures.

References

[1] Barhoumi, E. M., Charabi, Y., & Farhani, S. (2024). Detailed guide to machine learning techniques in signal processing. *Progress in Electronics and Communication Engineering, 2* (1), 39–47.

[2] Barla, I. B., Schupke, D. A., & Carle, G. (2012, July). Delay performance of resilient cloud services over networks. In *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications* (pp. 512-517). IEEE. https://doi.org/10.1109/ISPA.2012.75

[3] Chengsong, H. (2023). Time Delay Analysis of Deterministic Network 5G-A Terminals. *Adv. Comput. Signals Syst*, *7*, 57-61.

[4] Duan, Q. (2013, January). Modeling and delay analysis for converged network-cloud service provisioning systems. In *2013 International Conference on Computing, Networking and Communications (ICNC)* (pp. 66-70). IEEE. https://doi.org/10.1109/ICCNC.2013.6504055

[5] Duan, Q., & Vasilakos, A. V. (2016). Federated selection of network and cloud services for high-performance software-defined cloud computing. *International Journal of High-Performance Computing and Networking*, *9*(4), 316-327. https://doi.org/10.1504/IJHPCN.2016.077824

[6] Duan, Q., & Zheng, Z. (2014, May). Holistic modeling and performance evaluation for converged network-cloud service provisioning. In *2014 IEEE 28th International Conference on Advanced Information Networking and Applications* (pp. 978-984). IEEE. https://doi.org/10.1109/AINA.2014.10

[7] Erbad, A., Tayarani Najaran, M., & Krasic, C. (2010, February). Paceline: latency management through adaptive output. In *Proceedings of the first annual ACM SIGMM conference on Multimedia systems* (pp. 181-192). https://doi.org/10.1145/1730836.1730858

[8] Esmaeily, A., Kralevska, K., & Gligoroski, D. (2020, June). A cloud-based SDN/NFV testbed for end-to-end network slicing in 4G/5G. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)* (pp. 29-35). IEEE. https://doi.org/10.1109/NetSoft48620.2020.9165419

[9] Fountoulakis, E., Liao, Q., & Pappas, N. (2020). An end-to-end performance analysis for service chaining in a virtualized network. *IEEE Open Journal of the Communications Society*, *1*, 148-163. https://doi.org/10.1109/OJCOMS.2020.2966689

[10] Gajmal, Y. M., & Udayakumar, R. (2021). Blockchain-based access control and data sharing mechanism in cloud decentralized storage system. *Journal of web engineering*, *20*(5), 1359-1388. https://doi.org/10.13052/jwe1540-9589.2054

[11] Ghahramani, M. H., Zhou, M., & Hon, C. T. (2017). Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services. *IEEE/CAA Journal of Automatica Sinica*, *4*(1), 6-18. https://doi.org/10.1109/JAS.2017.7510313

[12] Goleva, R., Stainov, R., Kletnikov, N., Achkoski, J., Mirtchev, S., Ganchev, I., & Savov, A. (2017). Performance analysis of end-to-end sensor-to-cloud personal living platform. *Procedia computer science*, *113*, 615-620. https://doi.org/10.1016/j.procs.2017.08.309

[13] Hofmann, W., Lang, S., Reichardt, P., & Reggelin, T. (2022). A brief introduction to deploy Amazon Web Services for online discrete-event simulation. *Procedia Computer Science*, *200*, 386-393. https://doi.org/10.1016/j.procs.2022.01.237

[14] Huang, J., Duan, Q., Guo, S., Yan, Y., & Yu, S. (2015). Converged network-cloud service composition with end-to-end performance guarantee. *IEEE Transactions on Cloud Computing*, *6*(2), 545-557. https://doi.org/10.1109/TCC.2015.2491939

[15] Kanuparthy, P., Dai, Y., Pathak, S., Samal, S., Benson, T., Ghasemi, M., & Narayan, P. P. S. (2016). Ytrace: End-to-end performance diagnosis in large cloud and content providers. *arXiv preprint arXiv:1602.03273*. https://doi.org/10.48550/arXiv.1602.03273

[16] Khaleel, M. I., Safran, M., Alfarhood, S., & Zhu, M. (2023). Workflow scheduling scheme for optimized reliability and end-to-end delay control in cloud computing using AI-based modeling. *Mathematics*, *11*(20), 4334. https://doi.org/10.3390/math11204334

[17] Li, Y., Orgerie, A. C., Rodero, I., Amersho, B. L., Parashar, M., & Menaud, J. M. (2018). End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT. *Future Generation Computer Systems*, *87*, 667-678. https://doi.org/10.1016/j.future.2017.12.048

[18] Mohammed Hussain, V., Abdul Azeez Khan, A., Sathick, J., Raj, A., & Haja Alaudeen, A. (2025). Machine Learning Based Vehicle Traffic Patterns Prediction Model (ML-VTPM) With Mobile Crowd Sensing for Transportation Systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *16*(1), 1-25. https://doi.org/10.58346/JOWUA.2025.I1.001

[19] Nguyen, Q., Ghosh, P., & Krishnamachari, B. (2018, March). End-to-end network performance monitoring for dispersed computing. In *2018 International Conference on Computing, Networking and Communications (ICNC)* (pp. 707-711). IEEE. https://doi.org/10.1109/ICCNC.2018.8390359

[20] Oberle, K., Cherubini, D., & Cucinotta, T. (2013, September). End-to-end service quality for cloud applications. In *International Conference on Grid Economics and Business Models* (pp. 228-243). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-02414-1_17

[21] Pragadeswaran, S., Subha, N., Varunika, S., Moulishwar, P., Sanjay, R., Karthikeyan, P., ... & Vaasavathathaii, E. (2024). Energy Efficient Routing Protocol for Security Analysis Scheme Using Homomorphic Encryption. *Archives for Technical Sciences*, *31*(2), 148-158. https://doi.org/10.70102/afts.2024.1631.148

[22] Rahim, R. (2024). Scalable architectures for real-time data processing in IoT-enabled wireless sensor networks. *Journal of Wireless Sensor Networks and IoT*, *1*(1), 28-31. https://doi.org/10.31838/WSNIOT/01.01.07

[23] Ren, Y., Guo, S., Cao, B., & Qiu, X. (2023). End-to-end network SLA quality assurance for C-RAN: A closed-loop management method based on digital twin network. *IEEE Transactions on Mobile Computing*, *23*(5), 4405-4422. https://doi.org/10.1109/TMC.2023.3291012

[24] Sadulla, S. (2024). Techniques and applications for adaptive resource management in reconfigurable computing. *SCCTS Transactions on Reconfigurable Computing*, *1*(1), 6-10. https://doi.org/10.31838/RCC/01.01.02

[25] Santhosh, M., Kavitha, S., Keerthana, R., Suganya, L., & Krishnakumar, S. (2016). Electronic voting machine using internet. *International Journal of Communication and Computer Technologies*, *4*(2), 72-75.

[26] Scarpitta, C., Sidoretti, G., Mayer, A., Salsano, S., Abdelsalam, A., & Filsfils, C. (2023). High performance delay monitoring for SRv6-based SD-WANs. *IEEE Transactions on Network and Service Management*, *21*(1), 1067-1081. https://doi.org/10.1109/TNSM.2023.3300151

[27] Sikarwar, T. S., Chauhan, A. S., Jain, N., & Mathur, H. (2025). Application of Predictive Analytics in IOT Data Processing. *Indian Journal of Information Sources and Services, 15*(2), 340–348. https://doi.org/10.51983/ijiss-2025.IJISS.15.2.42

[28] Szigeti, T., Hattingh, C., Barton, R., & Briley Jr, K. (2013). *End-to-End QoS network design: Quality of Service for rich-media & cloud networks*. Cisco press.

[29] Takada, O., Ohshima, K., & Terada, M. (2013, January). End-to-end delay analysis method for tandem switches with variable bandwidth links. In *The International Conference on Information Networking 2013 (ICOIN)* (pp. 145-150). IEEE. https://doi.org/10.1109/ICOIN.2013.6496367

[30] Tang, L., Pu, Z., Hou, Q., Fang, D., & Chen, Q. (2024). Deterministic delay of digital twin-assisted end-to-end network slicing in industrial IoT via multiagent deep reinforcement learning. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2024.3476112

[31] Wu, C. Q., Lin, X., Yu, D., Xu, W., & Li, L. (2014). End-to-end delay minimization for scientific workflows in clouds under budget constraint. *IEEE Transactions on Cloud Computing*, *3*(2), 169-181. https://doi.org/10.1109/TCC.2014.2358220

[32] Ye, Q., Zhuang, W., Li, X., & Rao, J. (2018). End-to-end delay modeling for embedded VNF chains in 5G core networks. *IEEE Internet of Things Journal*, *6*(1), 692-704. https://doi.org/10.1109/JIOT.2018.2853708

[33] Yeo, M., & Jiang, L. (2024). Thermal and Fluid Systems: Analysis, Design, and Optimization. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *2*(1), 7-12.

[34] Zadeh, A. B., & Khalili, M. (2015). Security Solutions and challenges of cloud processing in line with service development. *International Academic Journal of Science and Engineering*, *2*(2), 175-181.

[35] Zheng, W., Hong, M., Ye, R., Fan, X., Liang, Y., Zhao, G., & Zimmermann, R. (2023). End-to-end delay modeling via leveraging competitive interaction among network flows. *IEEE Transactions on Network and Service Management*, *21*(2), 1634-1647. https://doi.org/10.1109/TNSM.2023.3340848

Authors Biography

Dr. Velpuru Muni Sekhar is an Associate Professor in the Department of Information Technology at Vardhaman College of Engineering, Hyderabad, Telangana, India. His research interests include artificial intelligence, machine learning, cloud computing, and data science. He has published numerous papers in reputed journals and conferences and is actively involved in research, teaching, and guiding students in emerging areas of information technology

Rishabh Vashisht is associated with the School of Engineering & Computing at Dev Bhoomi Uttarakhand University, Dehradun, India. His research interests include computer science, artificial intelligence, data analytics, and emerging computing technologies. He is actively engaged in academic research and contributes to initiatives that promote innovation, technical excellence, and practical learning in engineering and computing.

Rajat Saini is associated with the Centre of Research Impact and Outcome at Chitkara University, Rajpura, Punjab, India. His research interests include research analytics, innovation management, bibliometric studies, and impact assessment. He actively contributes to projects aimed at enhancing research quality, institutional visibility, and evidence-based evaluation of academic performance

Dr. Sarbeswar Hota is an Associate Professor in the Department of Computer Applications at Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India. His research interests include artificial intelligence, machine learning, data mining, and software engineering. With extensive academic and research experience, he has published several papers in reputed journals and conferences and is dedicated to fostering innovation and excellence in computer applications and emerging technologies

N. Vasantha Kumari is an Associate Professor in the Department of Computer Applications (DCA) at Presidency College, Bengaluru, Karnataka, India. Her research interests include artificial intelligence, data science, cloud computing, and software engineering. She has contributed to several academic publications and is actively involved in mentoring students and promoting innovation in computer science and application-oriented research.

Dr.R. Hannah Jessie Rani is an Assistant Professor in the Department of Electrical and Electronics Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Karnataka, India. Her research interests include renewable energy systems, power electronics, control systems, and smart grid technologies. She has published several research papers in reputed journals and conferences and is committed to advancing innovation and sustainable practices in electrical and electronics engineering