Frontiers in Computer Science and Artificial Intelligence
ISSN: 2798-8048
DOI: 10.32996/fcsai
Journal Homepage: www.al-kindipublisher.com/index.php/fcsai

A Comprehensive Survey of Scalable Content Delivery Architectures for Millions of Concurrent Users

Amit Mishra

Fox Corporation, Sr. Director of Engineering

Corresponding Author: Amit Mishra**, E-mail:** amitmishrajy@gmail.com

ABSTRACT

The rapid growth of online users and digital content has increased the demand for scalable content delivery architectures capable of supporting millions of concurrent users. These demands are met by Content Delivery Networks (CDNs), which use load balancing, globally dispersed servers, caching, and intelligent routing. The architectural underpinnings of content delivery, such as edge servers, Points of Presence (PoPs), communication models, and implementation techniques, are covered in this overview. In order to increase resource utilization and service availability, it also looks at scalability strategies including microservices design, cloud-based infrastructure, vertical scaling, and horizontal scaling. To maximize delivery performance, intelligent content distribution strategies such as AI-driven caching, predictive prefetching, dynamic content adaptation, and personalized recommendations are explored. Additionally, traffic management, congestion control, and machine learning-based resource allocation under varying workloads are examined in the study. Reliability, latency, throughput, reaction time, cache-hit ratio, and bandwidth use are all taken into account while evaluating performance. Notwithstanding these developments, monitoring overhead, diverse workloads, infrastructure expenses, security, and real-time resource optimization continue to present difficulties. In order to determine paths for creating effective, dependable, and economical content delivery systems on a broad scale, this analysis integrates existing architectural methods and operational issues.

KEYWORDS

Content Delivery Networks (CDNs), Scalability, Intelligent Traffic Management, AI-Driven Caching, Cloud Computing, Performance Evaluation, Load Balancing, Content Distribution.

ARTICLE INFORMATION

ACCEPTED: 17 October 2026 **PUBLISHED:** 17 November 2026 **DOI:** 10.32996/fcsai.2026.5.12.2

1. Introduction

In the contemporary digital era, the demand for multimedia content has witnessed an exponential surge due to the widespread adoption of high-speed internet, smart devices, and cloud-based applications [1]. From entertainment platforms such as Netflix, YouTube, and Amazon Prime Video to educational and corporate communication systems like Zoom, Coursera, and Microsoft Teams, the distribution of multimedia data has become a cornerstone of modern information technology infrastructure [2]. However, the massive and continuous increase in multimedia traffic has presented significant challenges related to scalability, latency, bandwidth consumption, and Quality of Experience (QoE) for end users. To address these challenges effectively, Cloud-Based Scalable Architectures for Multimedia Content Delivery Networks (CDNs) have emerged as a revolutionary paradigm, blending the strengths of cloud computing and traditional CDN infrastructures to deliver rich media content seamlessly across diverse geographical regions.

Content Delivery Networks (CDNs) are now critical enablers of the digital experience ecosystem and are vital in today's technologically enhanced world. CDNs distribute their content across numerous geographically diverse servers to connect users to the nearest servers, thus minimizing delays and response time. This decentralized approach reduces the load on origin servers and lets businesses maintain high performance even during peak traffic or in distant geolocations. Scalable system design is a crucial aspect of modern computing, enabling applications to efficiently handle increasing workloads, user demand, and data growth without compromising performance or reliability. In today's digital-first world, applications must remain responsive even during sudden spikes in traffic, making scalability a core requirement rather than an optional feature [3]. Traditional monolithic systems often struggle under such conditions, leading to the adoption of distributed architectures, cloud computing, and

Copyright: © 2026 the Author(s). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) 4.0 license (https://creativecommons.org/licenses/by/4.0/). Published by Al-Kindi Centre for Research and Development, London, United Kingdom.

microservices-based designs. Scalability ensures that systems can dynamically adjust resources based on demand while maintaining consistent performance, availability, and user experience.

CDNs are now more than just facilities that employ caching strategies. Rather, CDNs today feature sophisticated structures that take advantage of cutting-edge technologies like QUIC, HTTP/3, edge computing, and artificial intelligence, among others. These improve the dynamic, effective, and secure distribution of content, all of which are critical for businesses to maintain a competitive edge. The importance of CDNs as essential tools for developing dependable, readily scalable, and high-quality digital services has increased due to the development of mobility, IoT devices, and 5G networks. The purpose of this study is to determine the CDN's evolving nature, its existing architecture format, the different optimization techniques used in its current version, and potential future developments that might characterize this ground-breaking advancement.

1.1 Structure of the Paper

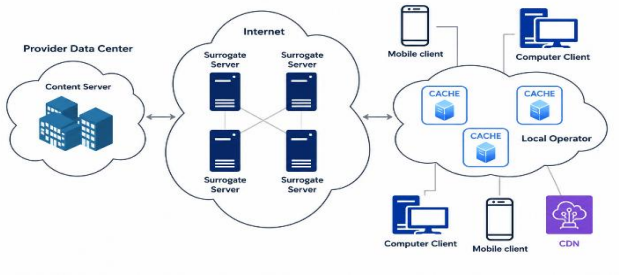
The paper is organized as follows: Section II presents content delivery architectures and scalability fundamentals. Section III discusses scalable content distribution and intelligent traffic management. Section IV examines performance evaluation, reliability, and operational challenges. Section V reviews recent literature and identifies research gaps. Finally, Section VI presents the conclusion and future research directions.

2. Content Delivery Architectures and Scalability

The architectures used for delivering content provide an effective method of distributing digital content to thousands of users concurrently using geographically dispersed servers, caching, and intelligent request routing. The scalable nature of the architecture is realized via the use of horizontal and vertical scaling, load balancing, cloud technology, and elastic computing.

2.1 Content Delivery Networks (CDNs)

The increasing number of Internet users and their growing demand for low-latency content delivery have led to the emergence of Content Delivery Networks (CDNs). The goal of a CDN is to locate the real responding server as near to the user as feasible [4]. The content provider provides a hierarchy of surrogate servers and cache servers to implement this concept, with the last tier often situated at the user network's edge.



2.2 Architecture of Current CDNs

Fig. 1 illustrates a CDN architecture connecting the provider data center, internet-based surrogate servers, local operator caches, and end-user devices to enable efficient content distribution, reduced latency, and scalable delivery.

2.2.1 Edge Servers

These act as geographically dispersed nodes that store cached material and respond to requests from users in the vicinity, representing the front-line infrastructure of CDN operations [5]. These servers are placed in a way that minimizes end consumers' geographical and network distances.

2.2.2 Points of Presence (PoPs)

These are important locations where several edge servers are clustered together to maximize coverage and performance in a certain region. PoPs are positioned strategically based on complex analyses of network architecture, user distribution, and traffic trends [6].

2.2.3 CDN Routing Mechanisms

The procedure makes use of sophisticated routing algorithms to guarantee that the request is directed to the best edge server for a user who requires content supplied via a CDN [7]. Analysis of CDN routing systems has demonstrated that sophisticated methods outperform simple geographical routing methods in terms of content delivery effectiveness.

2.2.4 Geographic Load Balancing

Geographic load balancing uses the user's location to send requests to the relevant edge servers. This approach uses complex mapping algorithms to link IP addresses to actual places. Studies have examined the precision of these systems.

2.2.5 Anycast Routing

It is a more sophisticated method in which several geographically dispersed servers use the same IP address. In CDN setups, empirical research on anycast implementation has shown that this technique can lower request routing latency compared to DNS-based redirection, especially for users on networks with less-than-ideal DNS resolver configurations.

2.2.6 Custom CDN Routing Algorithms

In order to make dynamic routing decisions, CDN providers have created unique routing algorithms that go beyond simple geographic or anycast approaches and take into account real-time network circumstances, server load, and user performance data [8]. These systems make use of vast telemetry networks that track internet performance over thousands of routes continually.

2.2.7 Caching System

The second infrastructure component that may be monitored in the present field is the caching system, research on which falls into two main categories [9]. The development of the entire caching system or a portion of it falls into the first category, while research on the effects of different parameters on the caching system falls into the second.

2.3 Communication in Content Delivery Networks

The communication schemes used in CDNs enhance the efficiency of content delivery, bandwidth usage, and network services. Peer-to-Peer (P2P) allows for resource sharing by the clients, whereas multicast sends content to multiple users at once. Anycast forwards requests to the nearest possible server.

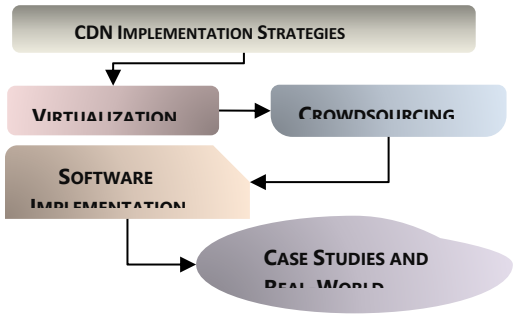
Peer-to-Peer: Several publications have examined client-level peer-to-peer (P2P) communication in CDNs since it allows clients to participate in the distribution of bandwidth.

Multicast: Another communication mechanism that CDN researchers take into consideration is multicast. However, the consumer network and the CDN network adopt the latter communication architecture.

Anycast: By forwarding the user's request to the nearest server, anycast in CDNs removes the requirement for expensive DNS support in smaller CDNs [10].

2.4 CDN Implementation Strategies

Several methods can be considered when implementing CDNs, including virtualization, software implementation, and crowdsourcing. Virtualization is useful for resource allocation, while software implementation allows several content providers to access cache servers. These strategies are shown in Fig. 2.



2.5 CDN Implementation Strategies

2.5.1 Virtualization

Enhancing resource utilization is the goal of this overlaying technique, which is based on on-demand resource distribution [11]. This method has been regarded as an efficient implementation strategy following its successful application in the deployment of software-defined networks and virtual servers.

2.5.2 Software Implementation

Several small content providers can overlay their CDN on top of a single set of cache servers supplied by the infrastructure provider thanks to a software-driven CDN implementation methodology that has been provided [12]. Additionally, there are issues with software-driven CDNs, including server-related issues.

2.5.3 Crowdsourcing

Crowdsourcing is an affordable way to deploy a CDN, but there are some difficult administrative issues that need to be addressed properly. In order to address the rivalry between content providers for CDN resources, researchers suggested an auction system in addition to a guiding framework [13]. Additionally, an effort has been made to leverage programmed caching strategies in a crowdsourced CDN.

2.5.4 Case Studies and Real-World Implementations

There are real-world case studies in the CDN sector that have already been published [14]. These case studies include reports about the passive information collection related to the deployment of YouTube's CDN. According to these reports, YouTube's CDN

has been viewed as a "black box," and it has been recognized how it handles a lot of traffic. In certain instances, the YouTube CDN has been proven to be unable to effectively handle traffic.

2.6 Scalability Models and Architecture

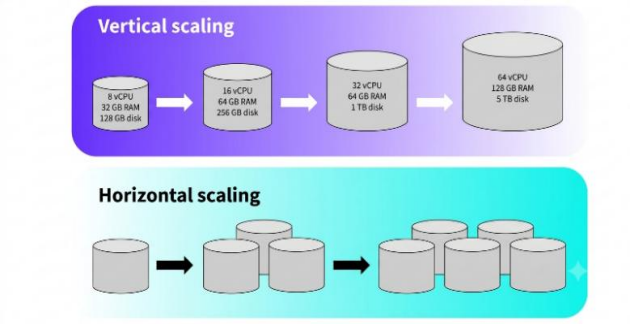
In the case of backend architecture, scalability is the capacity to manage additional operational demands related to user traffic, data volume, or calculation without lowering the output's overall efficiency. Scalability encompasses a system's ability to adapt to demands that change dynamically.

2.6.1 Vertical Scalability

It is also known as "scaling up," which is the process of increasing a machine's CPU, memory, or storage capacity to do additional work. Stronger hardware improves performance in this model (Fig. 2). In software design, vertical scaling is simple to do, but it is limited by the physical boundaries of a single machine.

2.6.2 Horizontal Scalability

Incorporating extra computers into the system to distribute the workload is known as scaling out. When the system may expand horizontally by adding more servers or nodes to a cluster, this approach is effective for distributed systems and web applications. The additional advantages of horizontal scaling in terms of flexibility and failure tolerance are shown in Fig. 3 since the system may continue to operate even if a component fails.



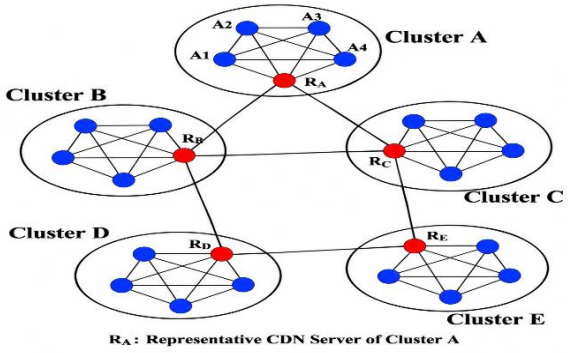
2.7 Vertical and Horizontal Scaling

2.7.1 Microservices Architecture

The software program is implemented as a collection of independent, self-contained, and fine-grained components that may be created separately, deployed, and tested together. With a database per microservice, these services are more controllable, testable, and maintainable since they employ lightweight protocols like HTTP and are matched to business capabilities.

2.7.2 Hierarchical Content Routing

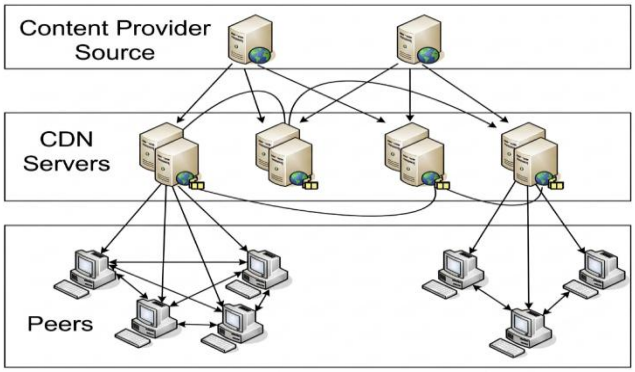
Hierarchical content routing organizes CDN servers into multiple clusters and uses representative nodes to coordinate inter-cluster communication. As seen in Fig. 4, this structure lowers routing overhead, enhances scalability, and permits effective content delivery through either self-organization or human setup. The CDN servers in the same cluster are in close proximity to one another.



2.8 Hierarchical Overlay Network Formed by CDN Servers

2.8.1 CDN-P2P Hybrid Network Architecture

This architecture combines CDN technology and P2P networks [15]. It is the type of architecture that reduces the stress on servers and optimizes the use of bandwidth.

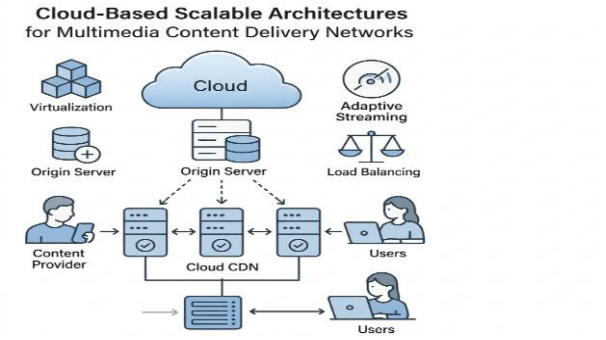


2.9 CDN-P2P Hybrid Network Architecture

Fig. 5 illustrates the CDN–P2P hybrid architecture, highlighting content provider servers, CDN caching servers, inter-CDN communication, peer-to-peer sharing, and distributed content delivery for improved scalability and bandwidth efficiency.

2.9.1 Cloud-Based Content Delivery Architecture

The cloud-based CDN solutions also encounter some challenges, such as latency differences based on the distance between data centers, vendor lock-in, compliance with data security laws, and energy expenditure for big data processing [16]. Related topics include green cloud computing, cross-cloud interoperability, and AI-based optimization.

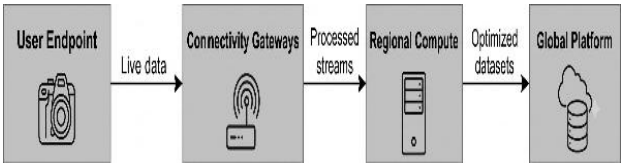


2.10 Cloud-Based Content Delivery Architecture

Fig. 6 illustrates a cloud-based scalable CDN architecture, highlighting virtualization, origin servers, edge servers, adaptive streaming, and load balancing to support efficient content distribution and reduce latency.

2.10.1 Edge Computing Architecture

Fig. 7 shows the overall framework used for edge computing [17]. Unlike cloud computing, which uses remote processing, edge computing uses localized processing on or near the device gathering data.



2.11 Edge Computing Architecture

Edge Device: The first piece is the actual edge device, such as an IoT device, camera, or sensor.

Edge Gateways: These are where data from multiple devices is aggregated. They also perform basic analytics and preprocessing, such as aggregation and format conversion.

Edge Servers: Local processing for real-time applications happens here, along with running containerized workloads or AI inference models.

Cloud or Data Center: Provides long-term storage, in-depth analytics, and model training, along with centralized management and orchestration.

Table I presents a comparative overview of CDN architectures, including traditional CDN, microservices, CDN–P2P hybrid, cloud-based, edge computing, and Multi-CDN architectures, highlighting their key components, functions, advantages, and challenges.

Overview of Content Delivery Network Architectures

Architecture Type	Key Components	Main Function	Advantages	Challenges
Traditional CDN Architecture	Origin servers, surrogate servers, hierarchical caches, edge servers	Delivers content from servers located closer to users	Reduced latency, faster access, efficient caching	Cache consistency, server placement, traffic distribution
Microservices Architecture	Independent services, APIs, lightweight protocols, service-specific databases	Separates CDN software functions into independently deployable services	Modularity, maintainability, flexibility, horizontal scalability	Service coordination, network overhead, monitoring complexity
CDN–P2P Hybrid Architecture	CDN servers, peer nodes, content providers, peer-to-peer connections	Combines centralized CDN delivery with distributed peer sharing	Reduced origin-server load, improved bandwidth utilization, scalability	Peer availability, security, content integrity, variable peer performance
Cloud-Based Content Delivery Architecture	Cloud infrastructure, origin servers, virtualized resources, load balancers, adaptive streaming	Uses cloud resources to distribute multimedia content dynamically	Elastic scalability, resource flexibility, efficient load distribution	Vendor lock-in, data compliance, energy consumption, distance-related latency
Edge Computing Architecture	Edge servers, edge caches, local processing nodes, connected devices	Processes and caches content near users to reduce dependence on remote cloud servers	Low latency, reduced backbone traffic, faster content access	Limited edge resources, distributed management, security and synchronization
Multi-CDN Architecture	Multiple CDN providers, traffic steering, monitoring systems, failover mechanisms	Distributes content across multiple CDN providers according to performance and availability	High availability, resilience, improved geographic coverage	Provider coordination, routing complexity, cost management, configuration overhead

3. Scalable Content Distribution and Intelligent Traffic Management

Intelligent caching, prediction, and server selection optimization in content distribution ensure effective content delivery. This is accomplished by using traffic management techniques that divide user requests among dispersed servers, hence lowering congestion and response times. Artificial intelligence makes it possible to forecast and distribute resources to increase efficiency by analyzing traffic patterns.

3.1 Traffic Management and Congestion Control

Content Delivery Networks encounter difficulties in coping with varying user demands, network congestion, and variable server loads. In this situation, traffic management necessitates sophisticated content delivery methods, resource management with regard to load, and dynamic request routing. Congestion analysis and mitigation are made easier by cloud computing, ML, and DL [18].

Adaptive Content Delivery in Connected User Environments: Adaptive delivery mechanisms utilize real-time user requests and network conditions to optimize content distribution, improve responsiveness, and overcome the limitations of traditional static delivery systems [19].

Traffic-Aware Content Distribution: Dynamic traffic management techniques analyze request density and workload variations to optimize routing decisions, balance server loads, reduce congestion, and maintain efficient content delivery under fluctuating demand [20].

Adaptive Content Delivery Using Cloud and Machine Learning: Cloud infrastructure and ML support dynamic resource allocation, workload prediction, congestion management, and anomaly detection, improving scalability and service reliability during traffic surges [21].

Deep Learning for Dynamic Content Delivery Optimization: DL and software-defined networking (SDN) can support intelligent traffic routing and resource scheduling by analyzing real-time network conditions and user demand, enhancing delivery efficiency for heterogeneous workloads [22].

3.2 AI-Driven Optimization of Content Delivery

AI-driven CDNs are not only improving online performance but also changing the future of the internet. The discussion also addresses potential adoption challenges, data privacy concerns, implementation costs, and the need for strong infrastructure, while offering ideas on how to overcome these hurdles to make full use of artificial intelligence in CDN contexts [23].

3.2.1 Intelligent Caching Mechanisms

Traditional caching policies such as Least Recently Used (LRU) and Least Frequently Used (LFU) are limited in handling the complex access patterns of modern content consumption [24]. AI-driven caching algorithms leverage behavioral, contextual, and performance data to improve cache allocation and hit rates.

3.2.2 Predictive Content Prefetching

Predictive prefetching aims to anticipate content requests based on user behavior, content relationships, and environmental context. Unlike caching, which responds to historical access, prefetching forecasts future demand, allowing CDNs to pre-load likely-to-be-requested content [25].

3.2.3 Dynamic Content Adaptation

This modifies content characteristics in real time based on device capabilities, user preferences, and network conditions [26]. This extends beyond bitrate adjustments to include resolution changes, format conversions, and content-aware compression.

3.2.4 Personalized Content Recommendations

Personalized content delivery enhances both user experience and CDN efficiency. AI-driven recommender systems predict individual preferences, enabling the strategic placement of relevant content closer to end-users [27]. User modeling incorporates behavioral signals, device usage, and temporal signals.

4. Performance Evaluation, Reliability, and Operational Challenges

Content delivery effectiveness is evaluated using metrics such as latency, throughput, response time, cache-hit ratio, and bandwidth consumption. Fault tolerance, server availability, and recovery techniques are used to increase service dependability. High traffic volumes, resource limitations, security concerns, and infrastructure expenses are a few of the operational challenges encountered during the process.

4.1 Scalability Challenges in Content Delivery Infrastructure

Content delivery infrastructure that scales encounters difficulties when dealing with variable workloads, resource allocation, and an increasing number of concurrent users. The scarcity of computing power, bandwidth, and storage space may affect its operation. Proactive scaling, load-sensitive resource allocation, and service-level tracking contribute to minimizing latency, using resources efficiently, maintaining reliability, and providing high-quality service even at times of high traffic.

4.1.1 Metric Scalability and Monitoring Overhead

As data centers develop to encompass tens or even hundreds of thousands of nodes, the process of data collection, aggregation, and analysis of demand metrics becomes an important issue. Although fine-grained monitoring is more accurate, it is excessively expensive in terms of computation, storage, and network overhead [28]. The infrastructure identified to satisfy particular needs aggravates this problem even further by increasing the number of measurable parameters.

4.1.2 Heterogeneity and Dynamic Workload Behavior

Modern data centers can be used for various workloads such as batch processing, interactive service delivery, and machine learning pipelines [29]. There are several different demands and sensitivities for these different workloads. The traditional metrics of demand have normally made the assumption of homogeneous resource usage, and this constrains their applicability in more heterogeneous settings using CPUs, GPUs, FPGAs, and hierarchical memories.

4.1.3 Accuracy and Responsiveness Trade-offs

The other essential scalability issue is the difficulty in balancing metric accuracy and responsiveness. Time-intensive analyses of historical data or sophisticated models used to estimate demand are often highly accurate but too slow for real-time decision-making [30]. On the other hand, lightweight metrics that react quickly may oversimplify workload behavior and result in inefficient provisioning or SLA violations.

4.2 Cloud Content Delivery Networks (CDNs)

CDNs have made a significant impact on how content is delivered via the Internet to end-users. Traditionally, content providers have relied on third-party CDNs to deliver their content to end-users [31]. It is difficult for content providers who either use third-party providers to augment their current delivery networks or fully depend on them to comprehend and track the performance of their service given the constantly shifting landscape of content types, such as going from standard definition video to high definition to full high definition.

4.2.1 Pay-as-you-Go CCDN Model

A cloud Content Delivery Network (CCDN) allows users to access delivered content using a pay-as-you-go model. Hence, it would be much more cost-effective than owning the physical infrastructure that is necessary for users to be part of a CDN.

4.2.2 Increased Points of Presence

The content is moved closer to users more easily in a CCDN system than in a traditional CDN due to the omnipresence of the cloud. The Cloud-based content delivery network can reduce transmission latency as it can rent operating resources from the cloud provider to increase the reach and visibility of the CDN on-demand.

4.2.3 CCDN Interoperability

CDN interoperability has become a strategically significant idea for content and service providers. CDN interoperability over the cloud enables content companies to accommodate mobile consumers and expand into new markets and geographical areas. For instance, providers can use existing regional cloud services to dynamically host surrogate servers rather than build infrastructure to serve a limited number of customers in Africa.

4.2.4 Support for a Variety of CCDNs

The cloud can support dynamic changes in load. This makes it easier for CDNs to accommodate a variety of applications with predictable and unpredictable bursting traffic, resource scaling, and rapid expansion and growth.

4.3 Scalable Content Delivery for Millions of Concurrent Users

Distributed server infrastructure, dynamic resource allocation, and effective request handling are necessary to deliver content to millions of users simultaneously at scale [32]. Caching, load balancing, and edge computing are used by content delivery networks to handle spikes in traffic. Increased throughput and reduced latency are the objectives of performance optimization, which also aims to maintain dependability and a consistent quality of experience during high demand periods.

Massive Concurrent User Demand: Sudden traffic surges, due to massive concurrent user demand, increase bandwidth consumption and server workloads.

High-Concurrency Request Processing: The server can serve many users simultaneously by processing requests concurrently. Distributed scheduling, asynchronous communication, and balanced workloads increase throughput and prevent processing bottlenecks.

Performance under Massive User Loads: Performance under massive user loads is dependent on latency, throughput, bandwidth utilization, and resource efficiency [33]. Regular monitoring helps to identify bottlenecks and maintain the same quality of delivery during peak traffic.

Reliability and Scalability during Traffic Surges: Reliable performance during traffic surges requires elastic resource provisioning, fault tolerance, and automated recovery mechanisms.

5. Literature Review

Recent research investigates the use of AI, cloud and edge computing, microservices, and hybrid CDN–P2P networks for scalable content distribution. These methods increase fault tolerance, lower bandwidth costs and rebuffering, and improve scalability. Real-time traffic optimization, resource allocation, and dependable delivery in the face of high concurrent demand, however, continue to be major obstacles.

H. Wang et al. (2026) introduced new cost challenges for Content Delivery Networks (CDNs). Unlike conventional Video on Demand (VoD) streaming, where CDNs can utilize the off-peak hours to preload video content on edge nodes, live streaming requires real-time delivery across CDN nodes before reaching end users, significantly increasing midgress bandwidth costs for service providers. Characteristics such as real-time content generation and consumption, strict requirements for service quality, and systems that need to scale with diverse resources to support the increase in large-scale users make it difficult to optimize live-streaming midgress traffic with traditional DNS-based and redirection-based methods [34].

V. Tian et al. (2026) present RLive, a robust delivery system that scales CDN capacity by integrating best-effort edge resources. RLive features a redundancy-free multi-source data plane to support reliable and cost-efficient live streaming, along with a multi-layer collaborative control plane that combines the global view with local adaptability for scalable user-to-node mapping. Deployed in ByteDance CDN to support large-scale live streaming services with hundreds of millions of daily viewers, RLive has tripled delivery capacity while reducing rebuffering events by 14.9–20.1% [35].

C. Henríquez et al. (2025) explored how Netflix's use of microservices architecture has improved its overall performance, scalability, and fault tolerance. The numerous architectural styles that Netflix used at different stages of its expansion are examined, emphasizing how these designs affected the company's capacity to adjust to a notable surge in users and content demand. Analyses of scholarly literature and industrial research are used to support the discussion of the main advantages and difficulties encountered throughout this shift [36].

M. Darwich et al. (2025) provide useful case studies and actual applications of cloud computing, edge computing, and artificial intelligence (AI) in video streaming. It looks at how these technologies have been used on platforms like Netflix, YouTube, and Amazon Prime Video to optimize video delivery, boost scalability, and improve user experiences. The case studies showcase cloud-based resource management, adaptive bitrate streaming, and AI-driven content recommendation systems. The chapter also highlights how edge computing may lower latency for real-time applications, such as interactive video experiences and live streaming [37].

D. Wei et al. (2024) presented ByteDance's cutting-edge hybrid video streaming technology, Swarm. Swarm skilfully combines the affordability of a decentralized P2P network with the resilience of a traditional CDN. Its main goal is to minimize traffic costs while offering consumers dependable streaming quality. Swarm uses a centralized control plane made up of a tracker cluster to manage a data plane that has a lot of edge residual resources in order to do this. Mapping clients to servers is another duty that the tracker assumes. Swarm uses a patented multipath parallel transmission technique for client-peer server communication in order to address performance differences between individual peer servers [38].

D. Iovescu et al. (2024) explored the field of dependable systems, with a particular emphasis on system architecture, software solutions, and architectural choices that enable cooperative work on shared text documents among several people in almost real time. It seeks to delve into the complexities of creating reliable and efficient document collaboration software, concentrating on comprehending the needs of such a system, the collaborative text editing principle, software architecture, technology stack selection, and tooling that can support such a system. The study describes how collaborative text editing software might benefit from such an architecture in terms of availability, flexibility, and scale in order to analyse the benefits and drawbacks of the suggested system [39].

R. Wadhwa et al. (2023) examined the use of event-driven microservices in conjunction with distributed data structures as a dependable choice for high-throughput commercial applications. The paper examines how microservices enable service deployment and scaling independently, enabling modular system design. Event-driven paradigms enhance system responsiveness by enabling asynchronous communication between scattered components. These tactics significantly improve system agility and fault tolerance when combined. Distributed data architecture, which decentralizes data processing and storage, significantly improves scalability [40].

Previous research has investigated scalable content distribution using cloud computing, edge computing, microservices, hybrid CDN–P2P systems, and CDNs. AI-driven video delivery, distributed system designs, enhanced delivery capacity, and bandwidth costs in live streaming have all been emphasized in research. But the main focus of these studies is on specific technologies or application scenarios. Existing methods do not provide a thorough comparative analysis of content delivery infrastructures under millions of concurrent users, as Table II summarizes. More research is especially needed to understand the combined consequences of latency, throughput, infrastructure cost, fault tolerance, resource utilization, and scalability. Furthermore, there is still a lack of attention to trade-offs between system dependability, operating costs, and delivery performance. In order to close these gaps, this review compares current architectures, assesses their advantages and disadvantages, and suggests future lines of inquiry for large-scale, dependable, and economical content distribution.

Summary of Key Studies on Scalable Content Delivery Architectures for Millions of Concurrent Users

Author & Year	Objective / Approach	Method / Technology	Key Findings	Advantages	Limitations
H. Wang et al. (2026)	Investigated cost challenges in large-scale live-streaming delivery.	CDN, edge nodes, DNS-based and redirection-based delivery.	Real-time delivery increases midgress bandwidth costs and complicates traffic optimization.	Highlights cost and scalability challenges in live streaming.	Traditional delivery methods struggle with diverse resources and strict QoS requirements.
V. Tian et al. (2026)	Introduced RLive for scalable and cost-efficient live streaming.	Multi-source data plane, collaborative control plane, edge resources.	Tripled delivery capacity and reduced rebuffering events by 14.9–20.1%.	Improves delivery capacity, reliability, and streaming quality.	Detailed implementation costs and broader comparative results are not provided in the supplied description.
C. Henríquez et al. (2025)	Examined Netflix's architectural evolution with increasing users and content demand.	Microservices architecture and alternative architectural styles.	Microservices enhanced scalability, fault tolerance, and overall performance.	Supports independent service development and adaptation to growing demand.	Transition challenges and architecture-specific performance measurements require further evaluation.

Author & Year	Objective / Approach	Method / Technology	Key Findings	Advantages	Limitations
M. Darwich et al. (2025)	Reviewed practical AI, cloud, and edge computing applications in video streaming.	AI-based recommendations, adaptive bitrate streaming, cloud resource management, edge computing.	These technologies optimize content delivery, scalability, and user experience.	Combines real-world applications with latency and resource optimization.	Quantitative comparisons across platforms are not specified in the supplied description.
D. Wei et al. (2024)	Introduced Swarm, a hybrid video-streaming system designed to reduce traffic costs.	Hybrid CDN–P2P architecture, centralized tracker, multipath parallel transmission.	Integrates CDN reliability with decentralized resource efficiency.	Improves streaming reliability and reduces traffic expenses.	Performance depends on peer availability and coordination; detailed numerical results are not provided.
D. Iovescu et al. (2024)	Explored dependable architectures for near-real-time collaborative text editing.	Distributed systems, collaborative editing software, availability and elasticity mechanisms.	Examined architectural decisions supporting collaboration, availability, and scaling.	Offers insights into dependable, multi-user software systems.	Focuses on document collaboration rather than large-scale content delivery.
R. Wadhwa et al. (2023)	Investigated high-throughput applications using event-driven microservices and distributed data architectures.	Event-driven communication, microservices, data partitioning, replication, eventual consistency.	Modular services and decentralized data management support scalability and fault tolerance.	Enables independent deployment, asynchronous processing, and flexible scaling.	Consistency management and distributed coordination can introduce additional complexity.

6. Conclusion and Future Work

Scalable content delivery architectures play a crucial role in distributing digital content efficiently to millions of concurrent users. This survey examined CDN architecture, including edge servers, Points of Presence, caching systems, routing mechanisms, and communication models. Horizontal and vertical scaling, cloud-based infrastructure, and microservices provide different approaches to managing increasing workloads and improving service availability. Furthermore, intelligent caching, predictive prefetching, dynamic content adaptation, and personalized recommendations support efficient content distribution and improved user experience. AI-driven traffic management and workload-aware resource allocation also contribute to reducing congestion and optimizing delivery performance. Performance evaluation through latency, throughput, response time, cache-hit ratio, and bandwidth utilization helps assess system effectiveness. However, monitoring overhead, heterogeneous workloads, infrastructure costs, security risks, and real-time resource allocation remain important challenges. Addressing these limitations requires coordinated architectural design, effective resource management, and reliable monitoring mechanisms. Overall, integrating scalable infrastructure with intelligent optimization techniques can support efficient, reliable, and cost-effective content delivery under rapidly changing user demands.

Future research should investigate adaptive AI-driven CDN architectures that combine predictive traffic analysis, edge computing, and elastic resource allocation. Further studies should evaluate performance under heterogeneous workloads and extreme traffic surges. Energy-efficient delivery, privacy-preserving optimization, cross-CDN interoperability, and real-world benchmarking also require attention to improve scalability, reliability, and cost-effectiveness.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] R. K. Kanneganti, "Impact of Custom Interface Widgets on User Efficiency in Content Navigation Platforms," Eastasouth J. Inf. Syst. Comput. Sci., vol. 1, no. 01, pp. 179–186, Aug. 2023, https://doi.org/10.58812/esiscs.v1i01.1121.

[2] A. Tyagi, "Optimizing digital experiences with content delivery networks: Architectures, performance strategies, and future trends," World J. Adv. Res. Rev., vol. 7, no. 2, pp. 401–417, Aug. 2022, https://doi.org/10.30574/wjarr.2020.7.2.0230.

[3] H. Umar, "A Review Of System Design For Scalable Applications," Int. J. Sci. Res. Eng. Trends, vol. 11, no. 2, pp. 1–6, 2025.

[4] B. Zolfaghari et al., "Content Delivery Networks," ACM Comput. Surv., vol. 53, no. 2, pp. 1–34, Mar. 2021, https://doi.org/10.1145/3380613.

[5] D. Munaretto, F. Giust, G. Kunzmann, and M. Zorzi, "Performance analysis of dynamic adaptive video streaming over mobile content delivery networks," in 2014 IEEE International Conference on Communications (ICC), IEEE, Jun. 2014, pp. 1053–1058. https://doi.org/10.1109/ICC.2014.6883460.

[6] S. P. Uppu, "Content Delivery Networks (CDNs) and live streaming: architecting scalable delivery for high-demand events," World J. Adv. Res. Rev., vol. 26, no. 2, pp. 2153–2164, May 2025, https://doi.org/10.30574/wjarr.2025.26.2.1849.

[7] K. Zarifis et al., "Diagnosing Path Inflation of Mobile Client Traffic," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8362, M. Faloutsos and A. Kuzmanovic, Eds., Cham: Springer International Publishing, 2014, pp. 23–33. https://doi.org/10.1007/978-3-319-04918-2_3.

[8] R. Meireles, J. Liu, and P. Steenkiste, "A study of HTTP/2's Server Push Performance Potential," pp. 1–8, Jul. 2022, https://doi.org/10.48550/arXiv.2207.05885.

[9] R. Afshoun, "Content Delivery Networks: Architectures, Security Challenges, and Future Directions," 2025. https://doi.org/10.13140/RG.2.2.24578.13763.

[10] H. Yang, H. Li, and Q. Wu, "An Integrating Unicast and Multicast Solution by Extending Content Delivery Network to Satellite," in 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, Jun. 2018, pp. 1414–1419. https://doi.org/10.1109/IWCMC.2018.8450502.

[11] H. Woo, S. Han, E. Heo, J. Kim, and S. Shin, "A Virtualized, Programmable Content Delivery Network," in 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, IEEE, Apr. 2014, pp. 159–168. https://doi.org/10.1109/MobileCloud.2014.32.

[12] H.-A. Tran, S. Souihi, D. Tran, and A. Mellouk, "MABRESE: A New Server Selection Method for Smart SDN-Based CDN Architecture," IEEE Commun. Lett., vol. 23, no. 6, pp. 1012–1015, Jun. 2019, https://doi.org/10.1109/LCOMM.2019.2907948.

[13] L. Sun, M. Ma, W. Hu, H. Pang, and Z. Wang, "Beyond 1 Million Nodes: A Crowdsourced Video Content Delivery Network," IEEE Multimed., vol. 24, no. 3, pp. 54–63, 2017, https://doi.org/10.1109/MMUL.2017.3051522.

[14] Z. Chang, L. Lei, Z. Zhou, S. Mao, and T. Ristaniemi, "Learn to Cache: Machine Learning for Network Edge Caching in the Big Data Era," IEEE Wirel. Commun., vol. 25, no. 3, pp. 28–35, 2018, https://doi.org/10.1109/MWC.2018.1700317.

[15] K. T. Nguyen and Y. H. Kim, "A CDN-P2P hybrid architecture with location/content awareness for live streaming services," KSII Trans. Internet Inf. Syst., vol. 5, no. 11, pp. 2143–2159, 2011, https://doi.org/10.3837/tiis.2011.11.015.

[16] Q. Jia, R. Xie, T. Huang, J. Liu, and Y. Liu, "The Collaboration for Content Delivery and Network Infrastructures: A Survey," IEEE Access, vol. 5, pp. 18088–18106, 2017, https://doi.org/10.1109/ACCESS.2017.2715824.

[17] F. C. Andriulo, M. Fiore, M. Mongiello, E. Traversa, and V. Zizzo, "Edge Computing and Cloud Computing for Internet of Things: A Review," Informatics, vol. 11, no. 4, p. 71, Sep. 2024, https://doi.org/10.3390/informatics11040071.

[18] A. Shanmugam, C. Fernandes, D. Sharma, and V. Upadhyaya, "Integrated Traffic Management System: A review," in Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing, New York, NY, USA: ACM, Aug. 2024, pp. 546–556. https://doi.org/10.1145/3675888.3676111.

[19] Z. Yao, Y. Jiang, B. Zhao, X. Luo, and B. Peng, "A dynamic optimization method for adaptive signal control in a connected vehicle environment," J. Intell. Transp. Syst., vol. 24, no. 2, pp. 184–200, 2020, https://doi.org/10.1080/15472450.2019.1643723.

[20] A. Shanmugam, C. Fernandes, D. Sharma, and V. Upadhyaya, "Integrated Traffic Management System: A review," in Proceedings of the 2024 Sixteenth International Conference on Contemporary Computing, in IC3-2024. New York, NY, USA, NY, USA, NY, USA: ACM, Aug. 2024, pp. 546–556. https://doi.org/10.1145/3675888.3676111.

[21] A. Yadav, V. More, N. Shinde, M. Nerurkar, and N. Sakhare, "Adaptive Traffic Management System Using IoT and Machine Learning," Int. J. Sci. Res. Sci. Eng. Technol., vol. 6, no. 1, pp. 216–229, Jan. 2019, https://doi.org/10.32628/IJSRSET196146.

[22] Z. Huang, "Reinforcement learning based adaptive control method for traffic lights in intelligent transportation," Alexandria Eng. J., vol. 106, pp. 381–391, Nov. 2024, https://doi.org/10.1016/j.aej.2024.07.046.

[23] G. Kambala, "Review on Accelerating Web Performance: The Role of AI-Driven Content Delivery Networks," 2024. [Online]. Available: https://www.researchgate.net/publication/388635144

[24] A. Kumar and M. N. Chauhan, "AI-Driven Optimization for Enhancing Performance, Efficiency, and Personalization in Content Delivery Networks," Int. J. Comput. Tech., vol. 12, no. 3, pp. 1–9, 2025, [Online]. Available: https://ijctjournal.org/

[25] H. Wang, H. Zhou, M. Lit, L. Zhao, and V. C. M. Leung, "Federated Distributed Deep Reinforcement Learning for Recommendation-enabled Edge Caching," in IEEE INFOCOM 2024 - IEEE Conference on Computer Communications

Workshops (INFOCOM WKSHPS), IEEE, May 2024, pp. 01–02. https://doi.org/10.1109/INFOCOMWKSHPS61880.2024.10620845.

[26] P. Yadav and S. Kar, "A cost-efficient content distribution optimization model for fog-based content delivery networks," J. Cloud Comput., vol. 13, no. 1, p. 141, Sep. 2024, https://doi.org/10.1186/s13677-024-00695-9.

[27] D. Sarkar, N. Rakesh, and K. K. Mishra, "Problems in Replica Server Placement (RSP) over Content Delivery Networks (CDN)," in Proceedings of the Sixth International Conference on Computer and Communication Technology 2015, New York, NY, USA: ACM, Sep. 2015, pp. 175–179. https://doi.org/10.1145/2818567.2818600.

[28] W. Wei, H. Gu, K. Wang, J. Li, X. Zhang, and N. Wang, "Multi-Dimensional Resource Allocation in Distributed Data Centers Using Deep Reinforcement Learning," IEEE Trans. Netw. Serv. Manag., vol. 20, no. 2, pp. 1817–1829, Jun. 2023, https://doi.org/10.1109/TNSM.2022.3213575.

[29] R. Ghosh and V. K. Naik, "Biting Off Safely More Than You Can Chew: Predictive Analytics for Resource Over-Commit in IaaS Cloud," in 2012 IEEE Fifth International Conference on Cloud Computing, IEEE, Jun. 2012, pp. 25–32. https://doi.org/10.1109/CLOUD.2012.131.

[30] N. K. Biswas, S. Banerjee, U. Ghosh, and U. Biswas, "Design of an energy efficient dynamic virtual machine consolidation model for smart cities in urban areas," Intell. Data Anal., vol. 27, no. 5, pp. 1409–1431, 2023, https://doi.org/10.3233/IDA-220754.

[31] M. Wang et al., "An overview of cloud based content delivery networks: Research dimensions and state-of-the-Art," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9070, pp. 131–158, 2015, https://doi.org/10.1007/978-3-662-46703-9_6.

[32] Y. Shin, H. Choi, Y. Nam, and E. Lee, "*PSO-Based Dynamic RSU Role Assignment Framework for Scalable and Reliable Content Delivery in VANETs.*," Sensors (Basel)., vol. 26, no. 5, p. 1555, Mar. 2026, https://doi.org/10.3390/s26051555.

[33] D. Yuan et al., "Understanding Operational CDN Live Streaming: A Measurement Study on Performance, Costs, and Enhancements," IEEE Trans. Circuits Syst. Video Technol., vol. 35, no. 7, pp. 7222–7238, 2025, https://doi.org/10.1109/TCSVT.2025.3538515.

[34] H. Wang et al., Medley: Optimizing Midgress Bandwidth for Commercial Live Streaming CDNs Optimizing Midgress Bandwidth for Commercial Live Streaming CDNs. USA: USENIX Association, 2026. [Online]. Available: https://www.usenix.org/conference/nsdi26/presentation/wang-haiping

[35] Y. Tian et al., "RLive: Robust Delivery System for Scaling Live Streaming Services," in Proceedings of the 21st European Conference on Computer Systems, in EUROSYS '26. New York, NY, USA: ACM, Apr. 2026, pp. 1–17. https://doi.org/10.1145/3767295.3769360.

[36] C. Henríquez, J. R. Valencia, and G. Sánchez, "Architectural Evolution from Monolithic to Microservices in Scalable Systems: A Case Study of Netflix," Prospect. ISSN-e 2216-1368, ISSN 1692-8261, Vol. 23, No. 1, 2025, vol. 23, no. 1, p. 12, 2025.

[37] M. Darwich and M. Bayoumi, "*Case Studies and Real-World Implementations of AI, Cloud, and Edge in Video Streaming,*" in Enhancing Video Streaming with AI, Cloud, and Edge Technologies: Optimization Techniques and Frameworks, Cham: Springer Nature Switzerland, 2025, pp. 289–310. https://doi.org/10.1007/978-3-031-84651-9_10.

[38] D. Wei et al., "Swarm: Cost-Efficient Video Content Distribution with a Peer-to-Peer System," Comput. Sci. > Netw. Internet Archit., pp. 1–14, Jan. 2024, https://doi.org/10.48550/arXiv.2401.15839.

[39] D. Iovescu and C. Tudose, "Real-Time Document Collaboration—System Architecture and Design," Appl. Sci., vol. 14, no. 18, p. 8356, Sep. 2024, https://doi.org/10.3390/app14188356.

[40] R. Wadhwa, "Designing Scalable Enterprise Systems Using Event-Driven Microservices and Distributed Data Architecture for High-Throughput Business Applications," Int. J. Comput. Eng. Technol., vol. 14, no. 3, pp. 323–337, 2023, https://doi.org/10.34218/ijcet_14_03_030.