Design and Optimization of Large-scale, Low-latency Multimedia Systems

Student Name:	Yulong Zhang
ID Number:	50006913
Thrust & Hub:	IoT, Information Hub
Supervisor:	Prof. Dirk Kutscher
Co-Supervisor:	Prof. Ying Cui

Dec. 2024

The Hong Kong University of Science and Technology (Guangzhou)

Content

Abstract

This paper presents a data-oriented overlay architecture for large-scale, low-latency multimedia systems, addressing key challenges in scalable content delivery and resource optimization. Our system features a hierarchical forwarding architecture where intermediate nodes make informed decisions about content delivery based on network metrics and downstream capabilities. This approach enables more efficient resource utilization compared to traditional client-side adaptation approaches. We propose two optimization models: a centralized Integer Linear Programming (ILP) model for achieving global optimality, and a distributed down-to-top model that decomposes optimization across forwarders. The distributed model reduces computational complexity while maintaining near-optimal performance. The system implements several key innovations: (1) capability-aware forwarding that considers client preferences and network conditions, (2) proactive quality probing for bandwidth utilization optimization. Simulation results demonstrate that our distributed model achieves up to 58.7% reduction in optimization solve time compared to centralized approaches while maintaining quality guarantees across diverse network conditions.

1 Introduction

The increasing demand for real-time multimedia applications, such as interactive streaming, virtual conferencing, and immersive virtual reality (VR), has posed new challenges for multimedia systems. These applications require the ability to handle millions of concurrent users while maintaining low latency (often under 20 ms) and high-quality content delivery [1]. However, existing systems often struggle to meet these demands due to inefficiencies in scalability, resource utilization, and architectural design. Conventional multimedia architectures based on client-server models and unicast delivery mechanisms face inherent limitations[2]. Traditional content delivery networks (CDNs) are often insufficient for latency-sensitive applications due to their static and hierarchical design, while Unicast approaches generate redundant network traffic [3], leading to inefficient bandwidth utilization. It is reported that virtual conferencing platforms frequently experience latency spikes and reduced performance as user numbers increase [4]. Recent research has sought to address these challenges through two main directions: 1) improving quality of experience (QoE) under network constraints and 2) developing novel

network protocols to enhance scalability and latency performance.

Optimizing Quality of Experience (QoE). QoE optimization methods aim to allocate resources effectively and adapt content delivery dynamically based on user and network conditions. Zhao et al. [5] proposed a one-to-many bitrate adaptation strategy to improve video quality while minimizing rebuffering time. However, this approach relies solely on lower-layer information, which limits its ability to account for dynamic network behaviors. Mao et al. [6] used deep reinforcement learning (DRL) to adapt video bitrate dynamically to changing network conditions. However, their method relies on heuristic training techniques, which lacks theoretical convergence guarantees. Twist [7] is a receiver-driven multi-site transport approach that uses receiver state information for global flow control to reduce flow competition. However, it faces fairness issues when competing for bandwidth with other protocols. TreeDN [8], a multicast-enabled content delivery network, reduces traffic by using Automatic Multicast Tunneling (AMT) to convert unicast streams into multicast streams at relay nodes. However, it inherits bottlenecks from traditional CDNs, such as the hierarchical caching model and inefficiencies at conversion points. BIER [9] eliminates multicast tree construction by embedding delivery information in packet headers, enabling scalable multicast. However, its domain-specific configuration requirements and limited support for real-time heterogeneous media environments constrain its applicability.

Protocols Innovations. Protocol-level innovations aim to address the inherent inefficiencies of traditional architectures. QUIC-based protocols [10], such as Twitch Warp [11] and Facebook RUSH [12], reduce connection setup latency and address TCP's head-of-line (HOL) blocking issue at the connection level. However, they still suffer from HOL blocking at the stream level. Each stream can still be held back by delayed segments from the same stream. Multipath TCP aggregates bandwidth across interfaces but retains TCP's limitations on heterogeneous paths [13]. Media over QUIC (MoQ) [14] introduces an application-level multicast overlay to support real-time media distribution, but it suffers from inefficiencies due to redundant unicast connections and separate control loops for each publisher-subscriber link. Data-oriented protocols aim to minimize latency through pull-based communication. ICN-RTC modifies WebRTC's media switching mechanism to improve scalability[15], but its multi-source synchronization features rely on complex consumer-side heuristics. ROBUST introduces NACK packets for payload information exchange [16] but suffers from inefficiencies in large-scale deployments. V-MAC, a content-centric MAC layer design, introduces a feedback-based

retransmission mechanism (DACK) to recover lost frames, minimizing latency and improving efficiency for applications sensitive to packet loss [17]. However, V-MAC focuses on minimizing loss rather than maximizing throughput or goodput, which can lead to inefficiencies for applications that require higher bandwidth utilization.

To address these limitations, we propose a data-oriented overlay architecture with distributed optimization. Our system integrates multicast at the network level and enables dynamic bandwidth allocation in multi-destination scenarios. The architecture features hierarchical forwarding where intermediate nodes make content delivery decisions based on network metrics and downstream capabilities. We develop a distributed optimization framework that decomposes the global resource allocation problem into sub-problems at each forwarding node, reducing computational complexity while maintaining performance in large-scale deployments.

The remainder of this paper is structured as follows: Section 2 provides a comprehensive background on modern multimedia applications, system architectures, and enabling technologies. Section 3 identifies key technical challenges in communication performance, scalability, and mobility. Section 4 presents the proposed optimization models and outlines future research directions. Finally, Section 9 details the research plan for implementing and validating these solutions.

2 Background

This section outlines the key applications, architectural foundations, and enabling technologies that underpin modern multimedia systems, emphasizing the challenges and opportunities in designing large-scale, low-latency solutions.

2.1 Modern Multimedia Applications

Multimedia applications today span a wide range of use cases [18, 19, 20], each with specific requirements that influence system design [21, 22]. Key categories include real-time collaboration, immersive environments, interactive media streaming, and adaptive content delivery. Real-time collaboration platforms, such as virtual meeting tools and remote workspaces, rely on low-latency communication to ensure seamless interaction [23]. For example, platforms like Zoom and Microsoft Mesh integrate audio, video, and shared content with stringent latency thresholds to maintain real-time responsiveness. These applications face scalability challenges as user numbers grow [4], particularly during large-scale events involving thousands of partici-

pants.

Immersive environments, including VR and augmented reality (AR) applications, demand ultra-low latencies and high-quality media delivery. VR platforms like Oculus and AR applications such as Pokémon GO [24] require seamless synchronization of virtual and real-world elements to preserve user immersion [25]. These applications often involve computationally intensive tasks, such as rendering 3D environments and delivering high-resolution, volumetric content [26]. Interactive media streaming, exemplified by live-streamed concerts or e-sports events, combines high-quality video distribution with real-time user interaction. Platforms hosting events with millions of participants, such as Fortnite's virtual concerts [27], highlight the need for efficient media synchronization and scalable architectures. These systems must balance bandwidth constraints and user experience across diverse network conditions. Across all these applications, maintaining QoS and QoE is critical [28]. Variations in network conditions, device capabilities, and user interactions necessitate adaptive mechanisms for media delivery, ensuring consistent performance and resource efficiency.

2.2 System Architecture

The architecture of multimedia systems plays a central role in determining their scalability, latency, and adaptability [29, 30]. While traditional models like client-server architectures remain prevalent, emerging paradigms, such as hybrid models and Information-Centric Networking (ICN), offer new opportunities to address current limitations [31].

Client-server architectures form the basis of most multimedia systems [4]. Centralized servers manage content storage and delivery, offering consistent quality for individual streams [32]. However, scalability becomes a bottleneck as user numbers increase, with server-side congestion and latency degrading overall performance. Content Delivery Networks (CDNs) address some of these challenges by caching content closer to users [33, 34], but their reliance on pre-encoded data limits real-time adaptability. Hybrid architectures, which combine elements of client-server and peer-to-peer (P2P) models, provide a more scalable alternative. These systems leverage edge computing to process and cache content closer to end-users, reducing latency and alleviating server loads [1]. For instance, edge servers can dynamically preprocess or transcode media, enabling applications like VR to deliver high-quality content without overwhelming central infrastructure. However, these systems require sophisticated resource management to ensure consistent performance [3]. Emerging paradigms, such as ICN, shift the focus from

host-based addressing to content-based delivery. By aggregating user requests for the same content and delivering it through multicast or in-network caching, ICN improves bandwidth efficiency and scalability [35]. This approach is particularly suited to applications with high concurrency, such as large-scale live-streaming or virtual conferences [36]. However, integrating ICN into existing infrastructure requires addressing compatibility and implementation challenges, including bandwidth demands [37].

2.3 Enabling Technologies

The success of large-scale, low-latency multimedia systems depends on several enabling technologies, including media encoding, rendering architectures, communication protocols, and dynamic quality adaptation mechanisms.

Media encoding and compression are essential for efficient content delivery. Advanced codecs like H.265 (HEVC) and AV1 reduce bandwidth requirements while preserving media quality [38]. Scalable video coding (SVC) further enables layered transmission, allowing systems to adapt quality based on network conditions [39]. However, new formats, such as volumetric video, present challenges with their high bandwidth demands and encoding latencies [37]. Real-time rendering technologies play a critical role in supporting resource-intensive applications. Rendering pipelines, which include input processing, content generation, and presentation, must meet stringent latency targets [40]. Techniques like collaborative rendering, where local devices handle lightweight tasks while remote servers process computationally intensive operations, help optimize performance [41]. These approaches are particularly relevant for VR and AR applications, where rendering delays directly impact user experience. Communication protocols underpin the real-time interaction and scalability of multimedia systems. Protocols like WebRTC and QUIC enable low-latency media transmission and dynamic adaptation to network conditions [42]. ICN-based communication further enhances scalability by supporting multicast and in-network caching, making it well-suited to high-concurrency applications [34]. Dynamic quality adaptation mechanisms ensure consistent QoS and QoE despite varying network conditions [28]. Techniques like adaptive bitrate streaming (ABR) and forwarder-based quality selection allow systems to balance performance and resource utilization [43]. These mechanisms enable multimedia systems to optimize delivery for diverse user requirements and device capabilities.

3 Technical Challenges

Large-scale, low-latency multimedia systems face numerous technical challenges stemming from the need to balance diverse requirements such as high throughput, low latency, scalability, and adaptability to dynamic environments [4]. These challenges span communication performance, mobility support, scalability, and end-system design, each contributing to the overall complexity of multimedia systems. This section explores these challenges in detail.

3.1 Communication Performance

Achieving efficient communication performance is important for ensuring smooth operation of multimedia systems, particularly for real-time and interactive applications [44]. Metrics such as latency, jitter, and throughput directly impact the delivery of multimedia content and the overall user experience. However, existing systems often face bottlenecks that hinder their ability to meet the stringent requirements of modern applications [45].

Latency remains a primary concern, particularly for motion-to-photon (MTP) applications such as VR, AR, and online gaming, where delays exceeding 20 milliseconds can lead to degraded user experiences or motion sickness [46]. The components of MTP latency include sensor detection, processing, rendering, and network transport. While advancements in hardware and local processing have reduced rendering latencies to 7-15 milliseconds [47], network-based systems still struggle with transport delays ranging from 50 milliseconds to over one second, depending on network conditions and congestion [45]. Jitter, the variability in latency, further exacerbates communication performance issues by causing frame drops and disruptions, especially in real-time streaming. For instance, jitter exceeding 33 milliseconds can negatively impact the playback of a 30 FPS video stream [48]. This problem is particularly pronounced in live-streamed events or interactive environments with varying network conditions and fluctuating user demands [49]. Throughput requirements in multimedia systems vary significantly based on resolution, compression, and frame rates [49]. High-resolution content such as 4K and 8K video, 360-degree video, or volumetric streaming demands substantial bandwidth [39], with peak throughput during the transmission of I-frames often exceeding average rates by a factor of 5-10 [50]. These peak bandwidth demands arise because I-frames do not benefit from inter-frame compression, posing challenges for bandwidth allocation and network resource optimization [51].

Existing transport protocols, such as TCP and QUIC, provide foundational solutions for communication but often fall short in addressing the demands of large-scale, low-latency systems. TCP suffers from slow congestion detection and recovery mechanisms, while QUIC, despite its improvements such as multiplexed streams and 0-RTT handshakes, struggles under high packet loss or frequent handovers [52]. Emerging protocols such as Low-Latency, Low-Loss, and Scalable Throughput (L4S) and deterministic networking (DetNet) aim to address these gaps [53]. L4S leverages Explicit Congestion Notification (ECN) to reduce latency [54], while DetNet reserves network resources to guarantee QoS for time-sensitive traffic [55]. However, these protocols remain experimental and require further refinement for real-world deployment in multimedia systems. Compression technologies play a key role in reducing the bandwidth required for multimedia delivery. Advanced codecs such as H.266 (Versatile Video Coding, VVC) and Video-based Point Cloud Compression (V-PCC) enable efficient handling of high-dimensional data [39]. Semantic compression techniques, which transmit only essential content, further optimize bandwidth utilization [56, 57]. However, the computational overhead of these advanced techniques poses challenges for energy-constrained devices and real-time applications [58]. Additionally, resource management strategies must address the disparity between average and peak bandwidth demands, particularly for dynamic, interactive applications [37]. Video streams encoded with layered techniques, such as Scalable Video Coding (SVC), enable adaptive quality adjustments, allowing systems to balance resource constraints with user QoE [39]. However, the dynamic nature of user demands and network conditions complicates the implementation of efficient resource allocation strategies.

3.2 Mobility Support

Mobility management is also a important aspect of large-scale, low-latency multimedia systems, ensuring uninterrupted service continuity as users move across heterogeneous network environments [4]. These systems face challenges in accommodating dynamic changes in connectivity, throughput, and latency due to user mobility, particularly when transitioning between different access technologies or network domains. Frequent handovers, caused by the deployment of small-cell ultra-dense networks (UDNs) in 5G and emerging 6G infrastructures, exacerbate mobility-related challenges [59]. The smaller cell radius and susceptibility of mmWave signals to fading and blockage result in frequent connectivity disruptions. These challenges are further intensified in applications requiring ultra-low latency and high reliability,

such as AR navigation or real-time VR gaming, where even brief interruptions during handovers can degrade the QoE [4]. For instance, in traditional TCP/IP-based systems, vertical handovers—where devices switch between different network technologies or operators—lead to changes in IP addresses and port numbers, causing session resets and delays in connection reestablishment.

Emerging technologies, including 6G networks and Space-Air-Ground Integrated Networks (SAGINs), aim to provide ubiquitous connectivity and seamless transitions across terrestrial and non-terrestrial networks [60]. While these advancements promise ultra-low latency (below 1 ms) and peak data rates of up to 1 Tbps, their integration introduces additional complexity [61, 62]. The increased diversity of access networks demands mobility management strategies that are both robust and adaptive to dynamic network conditions [63]. Handover mechanisms have been proposed to mitigate service disruptions. Horizontal handovers, which occur within the same network technology, can maintain consistent IP addresses, reducing disruptions [64]. Multi-connectivity architectures further enhance handover performance by enabling devices to connect to multiple base stations simultaneously [65], minimizing the latency associated with transitions. Advanced approaches such as random-access channel (RACH)-less handover schemes and make-before-break methodologies establish new connections before terminating existing ones, reducing service interruptions [66]. However, vertical handovers between different network technologies or operators remain more challenging, requiring coordination across network layers to maintain session continuity [67].

Protocols such as QUIC and its multipath extension (MPQUIC) provide innovative solutions for mobility management. Unlike TCP, QUIC employs connection identifiers instead of IP address-port tuples [52], allowing seamless session continuity during IP address changes. MPQUIC enhances performance by supporting multiple simultaneous paths [68] and enabling fast reconnections through zero round-trip time (0-RTT) handshakes [69]. However, standard congestion control mechanisms, such as those designed for TCP, may not perform well in mobility scenarios where packet loss arises from handovers rather than congestion [70]. Adaptive congestion control strategies that consider handover-induced disruptions are essential for optimizing performance in such scenarios. ICN offers a data-oriented alternative to mobility management by naming content rather than relying on host-based addressing. This approach inherently supports mobility by enabling stateful forwarding of named data packets [71], ensuring uninterrupted data access as users switch networks. Unlike traditional architectures, ICN

eliminates the need for session migration, allowing seamless access to the same content across networks. However, anchorless ICN architectures face challenges in managing simultaneous mobility of both endpoints [72], as the absence of fixed anchor points complicates path management and routing.

Moreover, gaps remain in achieving seamless mobility for large-scale multimedia systems. For example, transitioning between CDNs or edge nodes during vertical handovers often disrupts service continuity, as connections to the previous network cannot be maintained [67]. Real-time adaptation to new CDN environments and the efficient redistribution of content become critical in such scenarios. Additionally, the integration of diverse access networks in 6G, such as SAGINs, requires robust mechanisms to evaluate and select optimal paths for data transmission under varying network conditions [62].

3.3 Large-Scale Operation

Large-scale multimedia systems face challenges in managing scalability, communication efficiency, and state synchronization as the number of users and interactions increases. These systems must support millions of concurrent users while maintaining low latency and high throughput, which requires rethinking traditional architectural models and communication protocols [4]. One of the core challenges is scaling communication effectively. Many systems rely on centralized server-based architectures, where servers handle the management of virtual environments and user interactions [73]. While simple to implement, these architectures often become bottlenecks under increasing user loads. For instance, studies on social VR platforms have shown that latency and throughput demands scale linearly with the number of users, making it difficult to support more than 40 users in a single session [4]. This limitation arises from the exponential growth in interactions between users, which require frequent updates to shared states and media streams [74].

Existing communication models, such as unicast and inefficient multicast implementations, further exacerbate these challenges [4]. Unicast delivery methods, which send separate streams to each user, result in redundant data transmissions and high bandwidth consumption, particularly when users consume overlapping content such as shared fields of view in virtual environments. Multicast has been proposed as a solution to reduce bandwidth usage, especially for scenarios like live-streamed events or shared virtual experiences [75]. However, implementing multicast in traditional TCP/IP networks is complex due to the lack of acknowledgment

mechanisms at the link layer, difficulty in adapting to heterogeneous devices, and limited support for packet loss recovery. Caching and content distribution strategies have been extensively employed to improve scalability. CDNs replicate and cache data closer to end-users, reducing latency and offloading central servers [76]. However, traditional CDN models must adapt to the unique demands of multimedia systems, such as dynamic user interactions and rapidly changing content requirements. For example, in VR applications, caching strategies must account for user-specific fields of view (FoV) and prioritize high-quality content in regions of interest [77]. Advanced techniques like FoV-aware caching and deep learning-based cache placement models have been proposed to predict and optimize content delivery based on user behavior and interaction patterns [78]. These approaches can reduce latency by pre-loading high-priority tiles for immersive experiences. However, they introduce complexities in cache management, such as resource discovery, dynamic updates, and ensuring security [67].

Emerging protocols like Media over QUIC (MOQ) offer promising solutions to the challenges of large-scale operations [79]. MOQ extends the capabilities of QUIC by enabling efficient one-to-many communication and real-time media delivery [80]. By utilizing QUIC's inherent features, such as stream multiplexing and loss recovery, MOQ supports scalable content delivery while reducing latency. It also integrates well with CDN architectures by enabling in-network replication and prioritization of media packets. Applications like Twitch's Warp [11] and adaptations by Meta [81] demonstrate the potential of MOQ for interactive and large-scale multimedia applications. Despite these advancements, challenges persist in optimizing the routing of data paths, managing the complexity of overlay networks, and ensuring seamless integration with legacy systems [14]. ICN represents another approach to improving scalability [82]. Unlike traditional host-based networking, ICN focuses on delivering named content, enabling native multicast and efficient data dissemination. This approach eliminates the need for sender-based management of receivers, simplifying multi-destination delivery and reducing redundancy [83]. However, practical deployment of ICN faces hurdles, such as integrating with existing IP-based infrastructure, managing network topology changes, and ensuring compatibility with diverse application-layer protocols.

3.4 End System Architecture

The end system architecture of large-scale, low-latency multimedia systems should ensure efficient computation, rendering, and resource management [4]. These architectures must meet

the computational demands of rendering complex multimedia content, accommodate heterogeneous devices, and adapt to the dynamic nature of user interactions. Key challenges include optimizing rendering pipelines, addressing energy constraints of user devices, and ensuring efficient resource allocation across the network.

Rendering is one of the most resource-intensive processes in multimedia systems [4, 84]. It involves transforming raw data, such as 3D objects and volumetric video, into interactive visual environments. End-user devices, edge nodes, and cloud servers collectively contribute to rendering tasks, but the scalability of centralized architectures is limited [85]. For example, server-based systems like those in traditional VR environments often experience bottlenecks as the number of concurrent users increases. Decentralized approaches, such as peer-to-peer models, attempt to distribute computational tasks across multiple nodes, but this introduces communication overhead and system complexity. Efficient task partitioning and resource allocation are essential to balance these trade-offs [86].

Energy consumption is a critical constraint, particularly for mobile devices used in AR and VR applications. These devices often process high-quality content, leading to rapid battery depletion and heat generation [87, 88]. Interactive and remote rendering strategies offer solutions by offloading computational tasks to edge or cloud servers, reducing the load on local devices. However, such approaches increase the dependence on network performance, requiring high bandwidth and low latency to deliver interactive experiences [89]. Hybrid rendering methods, which divide tasks between local and remote resources, offer a compromise by enabling user devices to handle lightweight updates while offloading computationally intensive tasks to servers. These methods improve energy efficiency, extending device battery life by up to 50% [90], but require precise coordination to adapt to real-time variations in network and computational conditions.

Dynamic transcoding is another critical component of end system architecture, particularly for systems serving diverse devices and network conditions. By dynamically adjusting media content, transcoding ensures optimal quality for different device capabilities and bandwidth limitations [91, 92]. Edge-based transcoding frameworks reduce latency by performing these tasks closer to end users. For instance, metadata reuse during encoding processes minimizes computational costs while maintaining efficiency [93]. Despite their benefits, current transcoding solutions are primarily designed for video-on-demand applications and need adaptation for real-time, large-scale multimedia systems.

Modular system architectures have emerged as a solution to enhance adaptability and fault tolerance. Microservices-based designs decompose monolithic applications into smaller, independent services that can be dynamically orchestrated based on demand [94]. This modularity enables better scalability and task-specific optimizations, such as dynamic rendering or computational offloading. For multimedia systems, this approach provides the flexibility needed to handle diverse user requirements and application scenarios. In-network computing introduces additional opportunities to improve performance by moving computation closer to end users [95]. Tasks such as rendering, transcoding, and caching can be executed within the network, reducing latency and bandwidth usage. This approach is particularly beneficial for delay-sensitive applications, as it minimizes the need for centralized processing. However, challenges remain in balancing energy consumption and resource allocation across multiple network nodes, especially in environments with varying user demands and connectivity conditions [96]. Device-to-device (D2D) communication further enhances scalability by enabling nearby devices to share computational resources directly [97]. For instance, D2D-based caching schemes can prefetch and store frequently accessed content, reducing the load on network infrastructure during peak usage [98]. These methods improve overall efficiency but require sophisticated coordination mechanisms to manage resources effectively.

4 Research Vision and Current Research

The evolution of large-scale, low-latency multimedia systems demands a departure from traditional architectural paradigms to address the limitations of current approaches. Many existing systems struggle to meet the dual requirements of low-latency and scalability, largely due to centralized architectures, inefficient resource allocation, and an inability to fully leverage advanced network capabilities [99, 100]. Future architectures must be designed to support real-time, interactive communication for millions of concurrent users while addressing the challenges posed by heterogeneous devices, dynamic network conditions, and security requirements.

4.1 Next-Generation Architecture

To overcome current constraints, next-generation architectures must integrate the following key principles:

1. **Low-Latency and Scalable Communication:** Future systems must provide efficient me-

dia distribution capable of supporting both real-time interactions and large-scale deployments. Hierarchical and distributed communication models are needed to reduce reliance on centralized servers, mitigating bottlenecks and operational costs. For instance, edge computing and in-network processing can minimize latency by performing content adaptation and caching closer to users [101]. Additionally, multi-destination delivery methods, such as multicast or data-oriented approaches, can address bandwidth inefficiencies inherent in unicast protocols, particularly in applications with overlapping fields of view or shared content [100, 102].

2. **Dynamic and Modular Design:** A modular approach to system design is essential to ensure flexibility and adaptability. Functions such as rendering, transcoding, and resource management should be decomposed into smaller, independent components that can be dynamically orchestrated based on workload and network conditions. This modularity enables seamless task offloading to edge nodes or cloud servers, ensuring efficient use of computational and communication resources [99]. For instance, hybrid rendering techniques can distribute tasks between local devices and remote servers, adapting to changes in user demand and network capacity in real time [79].

3. **Secure and Adaptive Data Transformation:** Multimedia systems must support secure data transformation and adaptive computing to accommodate diverse user requirements and network conditions. This includes in-network processing for tasks such as dynamic transcoding and semantic compression, enabling systems to deliver personalized content at appropriate resolutions and quality levels [102, 103]. Secure communication protocols must be implemented to protect user data during storage, transmission, and processing without compromising performance. Future architectures should also incorporate trust models to ensure the integrity and confidentiality of user-generated content [101].

4. **Data-Oriented Communication Models:** Traditional host-centric networking approaches often introduce inefficiencies in multimedia systems, particularly for multi-user scenarios. Data-oriented models, such as ICN, provide a more efficient alternative by treating content as named data objects rather than relying on IP addresses. This enables native support for multicast delivery, in-network caching, and efficient resource discovery, reducing the overhead associated with traditional architectures [102]. For example, ICN can aggregate requests for the same content and deliver it once, significantly optimizing

bandwidth utilization in scenarios such as live streaming or virtual conferences [100].

5. **Resource Allocation and Scalability:** Future systems must incorporate intelligent resource allocation mechanisms that dynamically adapt to changing workloads and network conditions. This includes using predictive algorithms to prefetch and cache content, reducing latency and ensuring consistent QoE. Scalability can be further enhanced by employing distributed coordination techniques, allowing systems to scale seamlessly as the number of users grows [99, 102]. For instance, load balancing across edge nodes and dynamic reconfiguration of network paths can help maintain system performance under varying conditions.

4.2 Design Concepts

Future large-scale, low-latency multimedia systems require innovative design strategies to address the challenges of scalability, heterogeneity, and efficient resource utilization. Several design concepts can serve as a foundation for achieving these goals:

1. **Dynamic Function Offloading:** Splitting computational tasks between user devices, edge servers, and cloud resources can optimize resource usage and reduce latency. Hybrid rendering techniques, where computationally intensive tasks are offloaded to edge nodes or cloud infrastructure while lightweight updates are handled locally, are particularly effective [104, 79]. This approach not only reduces processing demands on end-user devices but also accommodates dynamic network conditions and varying device capabilities.

2. **Adaptive Data Distribution Frameworks:** Advanced data distribution mechanisms are essential to handle heterogeneous receiver groups and fluctuating network conditions. In-network caching and replication can reduce delays by bringing frequently accessed content closer to users [101]. Techniques such as dynamic transcoding at the edge enable the adaptation of content to match device capabilities, such as resolution or bandwidth constraints, ensuring a consistent quality of experience for all users [102].

3. **Modular System Architecture:** Decomposing monolithic applications into modular components improves flexibility and scalability. Independent components, connected through well-defined APIs, enable systems to dynamically adapt to changing workloads

and user demands [79]. This approach also facilitates the integration of emerging technologies, such as semantic compression, intelligent data filtering, and predictive analytics, into existing system architectures, ensuring continuous performance optimization.

4. **Data-Oriented Communication Models:** Shifting from traditional host-centric networking to data-oriented models can enhance efficiency in content delivery. Treating multimedia objects as named data units allows for more granular control of their distribution and retrieval [102]. For example, ICN natively supports multicast and in-network caching, optimizing bandwidth usage and reducing communication overhead in multi-user scenarios.

5. **Security-Aware System Design:** Protecting user data during storage, transmission, and processing is critical for large-scale multimedia systems. Robust encryption schemes, trust models, and secure protocols must be implemented without compromising performance [102]. In-network processing can enhance security by enabling real-time verification, transformation, and filtering of multimedia content within the network, reducing exposure to external threats [101].

6. **Context-Aware Resource Allocation:** Future systems must dynamically allocate resources based on real-time context, such as user behavior, application demands, and network conditions. Predictive algorithms and machine learning models can anticipate resource needs and optimize distribution, improving both scalability and efficiency [79, 102]. For example, viewport-aware streaming in virtual reality applications can prioritize high-resolution content for areas within the user's field of view, conserving bandwidth and computational resources.

5 System Design and Optimization

In this section, we present the implementation details and the optimization models used in our approach.

5.1 System Architecture

The proposed system implements a DAG-based architecture where network forwarders actively participate in content delivery optimization. Figure 1 illustrates the system architecture and communication flow between clients and forwarders. When requesting video content,

clients send Range-based Interests that include their supported resolution capabilities (e.g., [2K, 4K, 8K]) instead of requesting specific resolutions directly.

The forwarders serve as decision points in the network, processing these Range-based Interests through a two-step communication protocol. First, a client sends an Interest with the naming format `/ndn/video/<TitleID>/RangeInterest/chunk=<seq>` and includes its acceptable resolutions in the Parameters field. The forwarder then determines the optimal resolution based on network conditions and returns a NACK containing a recommended name (e.g., `/ndn/video/TitleA/2K/chunk=37`). Upon receiving this NACK, the client issues a standard NDN Interest using the recommended name to fetch the content.

This architecture enables forwarders to optimize content delivery by considering both global network conditions and aggregate user demands. The forwarders compute optimal configurations based on two key factors: the intersection of client capability sets and available upstream bandwidth. Unlike traditional client-side adaptation where clients adjust their quality independently, this approach allows forwarders to coordinate quality decisions across multiple users, leading to improved network resource utilization.

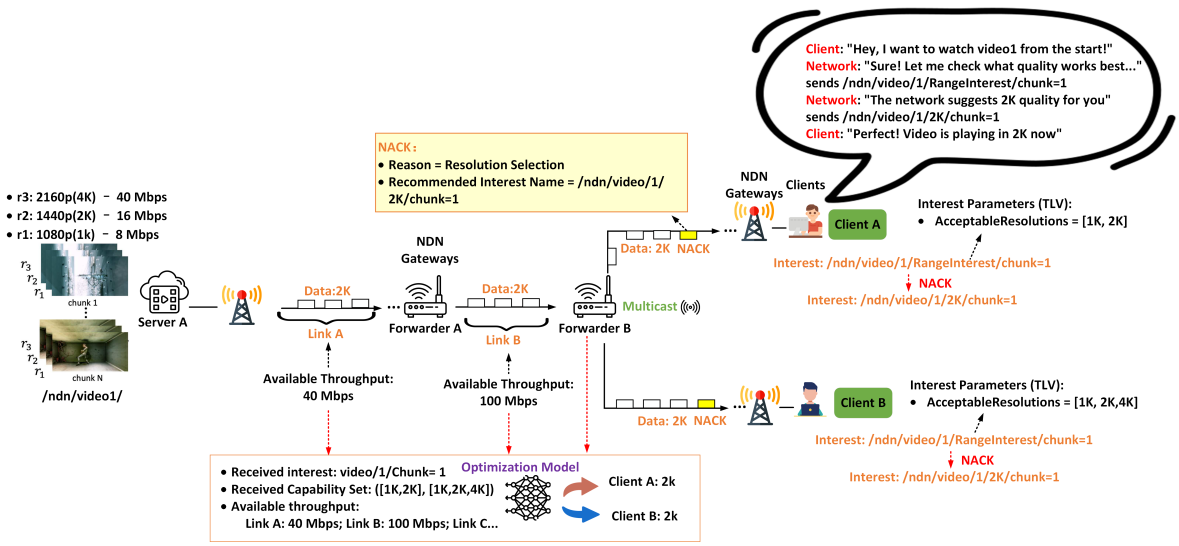

Figure 1: System Architecture. Resolution Selection through RangeInterest and NACK Mechanism. Forwarders optimize quality decisions based on client capabilities and available bandwidth.

5.2 Optimization Models

To evaluate resource allocation strategies, we propose and compare two optimization models: a centralized optimization model and a distributed model.

5.2.1 Centralized Optimization Model

Assumption The model assumes the existence of a **central node** that has access to global network information (including user demands, link states) and can make optimal resource allocation decisions from a global perspective.

$$\min \quad -\sum_{u\in\mathscr{U}} w_u \sum_{l\in\mathscr{L}} Q_l x_{u,l},$$

s.t.

$$x_{s,l} = 1, \quad s \in \mathscr{S},\, l \in \mathscr{L},$$

$$x_{u,l} = 0, \quad u \in \mathscr{U},\, l \in \mathscr{L} \setminus \mathscr{L}_u,$$

$$\sum_{l\in\mathscr{L}} x_{u,l} = 1, \quad u \in \mathscr{U},$$

$$y_{i,j,l} \le x_{i,l}, \quad (i,j) \in \mathscr{E},\, l \in \mathscr{L},$$

$$x_{i,l} \le \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}, \quad i \in \mathscr{U} \cup \mathscr{F},\, l \in \mathscr{L},$$

$$\sum_{l\in\mathscr{L}} B_l y_{i,j,l} \le C_{i,j}, \quad (i,j) \in \mathscr{E},$$

$$x_{i,l}, y_{i,j,l} \in \{0,1\}.$$

5.2.2 Sets and Parameters

- **Node Sets**

$$\mathscr{V} = \mathscr{S} \cup \mathscr{F} \cup \mathscr{U}$$

- $\mathscr{S}$: Set of servers (content source nodes), capable of providing all resolutions. - $\mathscr{F}$: Set of forwarders (intermediate nodes), capable of caching and forwarding content to downstream nodes. - $\mathscr{U}$: Set of users (terminal nodes), only consuming content.

Additionally, satisfying the following disjoint set relations:

$$\mathscr{S} \cap \mathscr{F} = \emptyset, \quad \mathscr{F} \cap \mathscr{U} = \emptyset, \quad \mathscr{S} \cap \mathscr{U} = \emptyset.$$

- **Edge Set and Graph**

$$\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$$

Graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is a directed acyclic graph, where $(i, j) \in \mathscr{E}$ represents allowed transmission from node i to node j.

  - **Incoming Edge Set**: For each node i:

$$\mathscr{I}_i = \{j \mid (j,i) \in \mathscr{E}\} \quad \text{(i.e., upstream of } i\text{)}$$

  - **Outgoing Edge Set**: For each node i:

$$\mathscr{O}_i = \{j \mid (i,j) \in \mathscr{E}\} \quad \text{(i.e., downstream of } i\text{)}$$

- **Resolution Set**

$$\mathscr{L} = \{1, 2, \ldots, l\}$$

Representing l different quality levels.

- **User Capability** For each user $u \in \mathscr{U}$, define

$$\mathscr{L}_u \subseteq \mathscr{L}$$

representing the set of quality levels available to user u.

- **Resolution Bandwidth Requirements**

$$B_l, \quad l \in \mathscr{L}$$

representing bandwidth consumption (in Mbps) of resolution l.

- **Link Maximum Capacity**

$$C_{i,j}, \quad (i,j) \in \mathscr{E}$$

representing available bandwidth limit (in Mbps) of edge (i, j).

- **Resolution Quality Score**

$$Q_l, \quad l \in \mathscr{L}$$

representing quality score of resolution l for user experience. To reflect diminishing marginal returns in user experience from resolution improvements, we model Q_l using the following logarithmic function:

$$Q_l = a + b \cdot \ln\left(\frac{l}{l_0}\right), \quad l \in \mathscr{L}$$

where l_0 represents baseline resolution, and coefficients a and b can be estimated through regression using subjective test data or statistical results from existing literature [**?**]. As l exceeds certain levels, the increment of $\ln(l/l_0)$ gradually slows.

- **User Priority Weights**

$$w_u, \quad u \in \mathscr{U}$$

representing importance coefficient of user u.

5.2.3 Decision Variables

- **Node-Resolution Selection Variables**

$$x_{i,l} \in \{0,1\}, \quad i \in \mathscr{V}, \, l \in \mathscr{L}$$

Meaning:

  - When $u \in \mathscr{U}$, $x_{u,l} = 1$ indicates user u selected resolution l.
  - When $f \in \mathscr{F}$, $x_{f,l} = 1$ indicates node f can obtain content at resolution l.
  - When $s \in \mathscr{S}$, $x_{s,l} = 1$ indicates server s provides resolution l.

- **Edge-Resolution Transmission Variables**

$$y_{i,j,l} \in \{0,1\}, \quad (i,j) \in \mathscr{E}, \, l \in \mathscr{L}$$

$y_{i,j,l} = 1$ indicates resolution l is transmitted on edge (i,j).

5.2.4 Objective Function

$$f(x) = \sum_{u \in \mathscr{U}} w_u \sum_{l \in \mathscr{L}} Q_l \, x_{u,l}$$

This is our objective function. The ultimate goal is to maximize this function, striving to serve more users with higher resolutions.

5.2.5 Constraints

Servers Fixed with All Resolutions

$$x_{s,l} = 1, \quad s \in \mathscr{S}, \, l \in \mathscr{L}$$

indicating server node s can provide all resolutions.

4.2 User Capability and Single Resolution Constraints

1. **Capability Limit**

$$x_{u,l} = 0, \quad u \in \mathscr{U}, \, l \in \mathscr{L} \setminus \mathscr{L}_u$$

   If resolution l is not in user u's available set, selection is prohibited.

2. **Single Resolution**

$$\sum_{l \in \mathscr{L}} x_{u,l} = 1, \quad u \in \mathscr{U}$$

   Each user selects at most one resolution.

Transmission Logic Constraints

1. **Content Availability**

$$y_{i,j,l} \leq x_{i,l}, \quad (i,j) \in \mathscr{E}, \, l \in \mathscr{L}$$

   Node i must first obtain resolution l ($x_{i,l} = 1$) before transmitting to downstream node j.

2. **Content Source**

$$x_{i,l} \leq \sum_{j:(j,i) \in \mathscr{E}} y_{j,i,l}, \quad i \in \mathscr{U} \cup \mathscr{F}, \, l \in \mathscr{L}$$

   For non-server nodes, obtaining any resolution must be through upstream transmission.

Bandwidth Capacity Constraints

$$\sum_{l \in \mathscr{L}} B_l \, y_{i,j,l} \leq C_{i,j}, \quad (i,j) \in \mathscr{E}$$

6 Distributed Lagrangian Model

In this section, we develop a distributed Lagrangian model for our resource allocation problem. In contrast to the centralized formulation, each node independently controls its own decision variables and the corresponding transmission variables toward its downstream nodes. In our model, the decision variables are partitioned as follows:

- $x_{i,l}$: the decision by node i (server, forwarder, or user) regarding whether to obtain or deliver resolution l.
- $y_{i,j,l}$: the decision by node i to transmit resolution l to downstream node j (with $j \in \mathscr{O}_i$).

The main coupling in the original problem is due to the *content source* constraint:

$$x_{i,l} \leq \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}, \quad i \in \mathscr{U} \cup \mathscr{F},\, l \in \mathscr{L},$$

which we relax by associating nonnegative Lagrange multipliers $\lambda_{i,l}$ (for non-server nodes only).

6.1 Model Summary

For ease of reference, we first summarize the centralized formulation that we intend to decompose:

$$\min \quad -\sum_{u\in\mathscr{U}} w_u \sum_{l\in\mathscr{L}} Q_l x_{u,l},$$

s.t.

$$x_{s,l} = 1, \quad s \in \mathscr{S},\, l \in \mathscr{L},$$

$$x_{u,l} = 0, \quad u \in \mathscr{U},\, l \in \mathscr{L} \setminus \mathscr{L}_u,$$

$$\sum_{l\in\mathscr{L}} x_{u,l} = 1, \quad u \in \mathscr{U},$$

$$y_{i,j,l} \leq x_{i,l}, \quad (i,j) \in \mathscr{E},\, l \in \mathscr{L},$$

$$x_{i,l} \leq \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}, \quad i \in \mathscr{U} \cup \mathscr{F},\, l \in \mathscr{L},$$

$$\sum_{l\in\mathscr{L}} B_l y_{i,j,l} \leq C_{i,j}, \quad (i,j) \in \mathscr{E},$$

$$x_{i,l}, y_{i,j,l} \in \{0,1\}.$$

6.2 Lagrangian Relaxation and Decomposition

Since each node i independently controls its own $x_{i,l}$ and its outgoing transmission variables $y_{i,j,l}$ (with $j \in \mathscr{O}_i$), we relax the coupling constraint

$$x_{i,l} \leq \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}$$

by introducing the multipliers $\lambda_{i,l} \geq 0$ (for $i \in \mathscr{U} \cup \mathscr{F}$). The resulting Lagrangian function is given by

$$\begin{aligned}\mathscr{L}(x,y,\lambda) &= -\sum_{u\in\mathscr{U}} w_u \sum_{l\in\mathscr{L}} Q_l x_{u,l} + \sum_{i\in(\mathscr{U}\cup\mathscr{F})} \sum_{l\in\mathscr{L}} \lambda_{i,l}\Big(x_{i,l} - \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}\Big) \\ &= \sum_{u\in\mathscr{U}} \sum_{l\in\mathscr{L}} \Big[(\lambda_{u,l} - w_u Q_l) x_{u,l}\Big] + \sum_{f\in\mathscr{F}} \sum_{l\in\mathscr{L}} \lambda_{f,l} x_{f,l} - \sum_{i\in(\mathscr{U}\cup\mathscr{F})} \sum_{j:(j,i)\in\mathscr{E}} \sum_{l\in\mathscr{L}} \lambda_{i,l} y_{j,i,l}\,.\end{aligned}$$

Note that for any edge (j,i) (with $i \in \mathscr{U} \cup \mathscr{F}$), the decision variable $y_{j,i,l}$ is controlled by the upstream node j, and it appears in the Lagrangian with coefficient $-\lambda_{i,l}$.

6.3 Decomposed Node Subproblems

We now decompose the global Lagrangian into three types of subproblems based on node roles.

6.3.1 Server Subproblem ($s \in \mathscr{S}$)

For server nodes, the content availability decision is fixed, i.e., $x_{s,l} = 1$ for all $l \in \mathscr{L}$. However, the server still determines its transmission decisions $y_{s,i,l}$ for all downstream nodes $i \in \mathscr{O}_s \cap (\mathscr{U} \cup \mathscr{F})$. Its objective is to account for the corresponding downstream multiplier contributions:

$$\begin{aligned}\mathscr{P}_s: \quad \min_{y_{s,\cdot,\cdot}} \quad & -\sum_{i\in\mathscr{O}_s\cap(\mathscr{U}\cup\mathscr{F})} \sum_{l\in\mathscr{L}} \lambda_{i,l} y_{s,i,l}, \\ \text{s.t.} \quad & y_{s,i,l} \in \{0,1\}, \quad \forall i \in \mathscr{O}_s,\, l \in \mathscr{L}, \\ & \sum_{l\in\mathscr{L}} B_l y_{s,i,l} \leq C_{s,i}, \quad \forall i \in \mathscr{O}_s.\end{aligned}$$

6.3.2 Forwarder Subproblem ($f \in \mathscr{F}$)

A forwarder node f makes two sets of decisions:

- The resolution acquisition decision $x_{f,l}$ (i.e., whether node f obtains resolution l).
- The transmission decisions $y_{f,k,l}$ for downstream nodes $k \in \mathscr{O}_f$.

Its Lagrangian contribution consists of a term $\lambda_{f,l} x_{f,l}$ for its own resolution decision and a term $-\lambda_{k,l} y_{f,k,l}$ for each downstream transmission. The forwarder subproblem is formulated as:

$$\begin{aligned}
\mathscr{P}_f: \quad & \min_{x_{f,\cdot}, y_{f,\cdot,\cdot}} \quad \sum_{l \in \mathscr{L}} \lambda_{f,l} x_{f,l} - \sum_{k \in \mathscr{O}_f \cap (\mathscr{U} \cup \mathscr{F})} \sum_{l \in \mathscr{L}} \lambda_{k,l} y_{f,k,l}, \\
\text{s.t.} \quad & y_{f,k,l} \le x_{f,l}, \quad \forall k \in \mathscr{O}_f,\ l \in \mathscr{L}, \\
& \sum_{l \in \mathscr{L}} B_l y_{f,k,l} \le C_{f,k}, \quad \forall k \in \mathscr{O}_f, \\
& x_{f,l} \in \{0,1\}, \quad y_{f,k,l} \in \{0,1\}.
\end{aligned}$$

6.3.3 User Subproblem ($u \in \mathscr{U}$)

A user node u only selects a resolution l (i.e., it decides $x_{u,l}$) and its objective is affected by both the user payoff and the Lagrangian penalty:

$$\begin{aligned}
\mathscr{P}_u: \quad & \min_{x_{u,\cdot}} \quad \sum_{l \in \mathscr{L}} \Big(\lambda_{u,l} - w_u Q_l \Big) x_{u,l}, \\
\text{s.t.} \quad & \sum_{l \in \mathscr{L}} x_{u,l} = 1, \\
& x_{u,l} \in \{0,1\}, \quad x_{u,l} = 0,\ \forall l \notin \mathscr{L}_u.
\end{aligned}$$

6.4 Dual Updates and Distributed Algorithm

Once the node subproblems are solved locally, each node exchanges the necessary information (e.g., each node i receives the transmission decisions $y_{j,i,l}$ from its upstream nodes $j \in \mathscr{I}_i$). The dual multipliers are then updated via a subgradient method. Specifically, for each $i \in \mathscr{U} \cup \mathscr{F}$ and $l \in \mathscr{L}$, we update:

$$\lambda_{i,l}^{(t+1)} = \max\Big\{0,\ \lambda_{i,l}^{(t)} + \theta^{(t)} \Big[x_{i,l}^{(t)} - \sum_{j:(j,i) \in \mathscr{E}} y_{j,i,l}^{(t)} \Big] \Big\},$$

where $\theta^{(t)}$ is a step-size parameter at iteration t.

A high-level pseudocode of the distributed algorithm is given below.

Algorithm 1 Distributed Lagrangian Subgradient Method

Input: initial multipliers $\lambda^{(0)} \geq 0$, step-size sequence $\{\theta^{(t)}\}$, iteration limit $T_{\max}$

Output: approximate solution (x, y) and multipliers λ

$t \leftarrow 0$

while $t < T_{\max}$ and not converged **do**

    (1) In parallel, solve node subproblems:

    for all server node $s \in \mathscr{S}$ **do**

        Solve $\mathscr{P}_s$ to obtain $\{y_{s,i,l}^{(t)}\}$

    end for

    for all forwarder node $f \in \mathscr{F}$ **do**

        Solve $\mathscr{P}_f$ to obtain $\{x_{f,l}^{(t)}, y_{f,k,l}^{(t)}\}$

    end for

    for all user node $u \in \mathscr{U}$ **do**

        Solve $\mathscr{P}_u$ to obtain $\{x_{u,l}^{(t)}\}$

    end for

    (2) Exchange necessary information: each node i receives $\{y_{j,i,l}^{(t)}\}$ from upstream nodes j

    (3) Update dual multipliers:

    for all $i \in \mathscr{U} \cup \mathscr{F}$ and $l \in \mathscr{L}$ **do**

        $\lambda_{i,l}^{(t+1)} = \max\Big\{0,\ \lambda_{i,l}^{(t)} + \theta^{(t)} \Big[x_{i,l}^{(t)} - \sum_{j:(j,i)\in\mathscr{E}} y_{j,i,l}^{(t)}\Big]\Big\}$

    end for

    $t \leftarrow t + 1$

end while

return $(x^{(t)}, y^{(t)})$ and $\lambda^{(t)}$

6.5 Discussion

By decomposing the global problem into subproblems that can be solved locally and then coordinated via dual updates, the proposed distributed Lagrangian model better reflects the inherent information flow in distributed systems. Each node's decision process is localized, and only limited information (i.e., the transmission decisions from upstream nodes) is exchanged during each iteration. Although the direct application of Lagrangian relaxation to binary variables may lead to a continuous relaxed solution, a subsequent integer mapping or heuristic

correction is typically required to obtain an integer-feasible solution.

6.6 Complexity Analysis and Comparison

6.6.1 Centralized Model

The centralized model solves a global optimization problem, where the complexity is determined by the number of variables and constraints. The overall complexity is:

$$O(2^V \cdot \text{poly}(C)),$$

where:

$$\text{Variables: } V = |\mathscr{V}| \cdot |\mathscr{L}| + |\mathscr{E}| \cdot |\mathscr{L}|,$$

$$\text{Constraints: } C = |\mathscr{S}| \cdot |\mathscr{L}| + |\mathscr{U}| \cdot |\mathscr{L}| + |\mathscr{U}| + |\mathscr{E}| \cdot |\mathscr{L}| + (|\mathscr{U}| + |\mathscr{F}|) \cdot |\mathscr{L}| + |\mathscr{E}|,$$

with the following definitions:

- $|\mathscr{V}|$: Total number of nodes, including servers ($\mathscr{S}$), forwarders ($\mathscr{F}$), and users ($\mathscr{U}$).
- $|\mathscr{E}|$: Total number of edges (connections between nodes).
- $|\mathscr{L}|$: Number of resolution levels or quality levels.

Key Complexity Sources:

1. **Number of Variables:** Includes content ownership variables ($x_{i,l}$) and transmission variables ($y_{i,j,l}$).
2. **Number of Constraints:** Covers server constraints, user resolution selection, transmission logic, and bandwidth limitations.
3. **Exponential Complexity:** The binary nature of $x_{i,l}$ and $y_{i,j,l}$ makes the problem NP-hard.

Limitations:

- **Global Scale:** Complexity grows exponentially with network size ($|\mathscr{V}|$ and $|\mathscr{E}|$).
- **Impractical for Large Networks:** The centralized model is infeasible for real-time applications in large-scale systems.

6.6.2 Distributed Model

The distributed model divides the global problem into local subproblems, reducing the complexity for individual nodes. The overall complexity can be expressed as:

$$O\left(\sum_{r\in\{\mathscr{S},\mathscr{F},\mathscr{U}\}} \text{iter}_{\max}\cdot C_r\right),$$

where:

- $r \in \mathscr{S},\mathscr{F},\mathscr{U}$: Represents servers, forwarders, and users.
- $\text{iter}_{\max}$: Maximum number of iterations required for convergence.
- C_r: Local complexity of each node.

Local Complexity by Node Type:

1. **Server Nodes ($s \in \mathscr{S}$)**:

$$C_s = O(|\mathscr{O}_s|\cdot|\mathscr{L}|),$$

where $|\mathscr{O}_s|$ is the number of outgoing links for server s.

2. **Forwarder Nodes ($f \in \mathscr{F}$)**:

$$C_f = O(|\mathscr{L}|\cdot(|\mathscr{I}_f|+|\mathscr{O}_f|)),$$

where $|\mathscr{I}_f|$ and $|\mathscr{O}_f|$ are the numbers of incoming and outgoing links for forwarder f, respectively.

3. **User Nodes ($u \in \mathscr{U}$)**:

$$C_u = O(|\mathscr{L}_u|\cdot|\mathscr{I}_u|),$$

where $|\mathscr{L}_u|$ is the set of available resolutions for user u, and $|\mathscr{I}_u|$ is the number of incoming links.

Communication Overhead:

$$O\left(\sum_{i\in\mathscr{V}} |\mathscr{I}_i|\cdot|\mathscr{L}|\right),$$

Each node communicates with its upstream neighbors to exchange resolution transmission variables $y_{j,i,l}$.

6.6.3 Comparison of Centralized and Distributed Models

Table 2: Comparison of Centralized and Distributed Models

Aspect	**Centralized Model**	**Distributed Model**
Variables (V)	Global: $\lvert\mathscr{V}\rvert \cdot \lvert\mathscr{L}\rvert + \lvert\mathscr{E}\rvert \cdot \lvert\mathscr{L}\rvert$	Local: depends on $\lvert\mathscr{I}_i\rvert + \lvert\mathscr{O}_i\rvert$
Constraints (C)	Global constraints on all nodes and links	Local constraints on neighbors and l
Complexity	$O(2^V \cdot \text{poly}(C))$	$O\left(\sum_r \text{iter}_{\max} \cdot C_r\right)$
Parallelism	Not supported	Fully parallelizable
Scalability	Poor	Excellent for large-scale network
Communication Cost	None	Exists, but grows moderately with netw

6.6.4 Conclusion

The centralized model is suitable for small-scale systems, but its exponential complexity makes it infeasible for real-time applications in large-scale networks. The distributed model, by leveraging local computations and parallelism, reduces per-node complexity and enables scalability. While it introduces communication overhead and requires iterative updates, it is highly practical for large-scale distributed networks.

7 Implementation

7.1 Throughput Measurement

Link throughput measurement uses Interest packets to probe bandwidth availability, similar to DASH bitrate testing. This approach avoids RTT measurement issues in ICN environments where:

- Data may traverse multiple paths or come from multiple sources.
- Interest aggregation in Pending Interest Tables affects delay measurements.
- Cache dynamics can cause variations in retrieval times.
- Packet loss can trigger false timeouts.

The system handles these challenges through periodic probing and maintains stable measurements even with path changes or cache exhaustion.

7.2 Negotiation-Based Protocol Designs

7.2.1 Protocol Overview

The proposed protocol enables efficient video resolution selection through a two-phase process: initial negotiation and periodic optimization. When a client first joins, it communicates its resolution capabilities through a RangeInterest mechanism. Subsequently, the network periodically optimizes resolution assignments based on stored client capabilities and network conditions, using NACK messages to notify clients of resolution changes.

7.2.2 RangeInterest and NACK Mechanism

RangeInterest Format A client initiates video streaming by sending a RangeInterest:

`/ndn/video/<TitleID>/RangeInterest/chunk=`$\langle seq \rangle$

The Interest's Parameters field contains `AcceptableResolutions`, listing the client's supported resolutions (e.g., [2K, 4K, 8K]). This information is stored by the forwarder for future optimization cycles.

NACK Response When resolution adjustment is needed, the forwarder responds with a NACK:

```
Reason: Resolution Selection
Recommended Name: /ndn/video/<TitleID>/<Res>/chunk=<seq>
```

The client then issues a standard NDN Interest using the recommended name to fetch the content.

7.2.3 Resolution Optimization

Periodic Optimization Following DASH-based streaming systems (e.g., DASH-Netflix (4-second assessment frequency) or DASH-YouTube (5 seconds)), the network performs resolution optimization every T=4 seconds using the stored client resolution capabilities. The forwarder maintains a mapping of active clients and their capabilities:

`map{capabilities, current_resolution}`

When optimizing, they will evaluate current network conditions and Updates the resolution assignment table:

$$\text{AssignmentTable}[u] = \text{Res}_u^*, \quad u \in \mathscr{U}$$

On New User Arrival When a new user issues its first `RangeInterest`, the forwarder will perform an *immediate* global optimization. If this optimization affects existing users' assignments (e.g., the new user's arrival changes the bandwidth share), the forwarder sends `NACKs` to only those users requiring resolution adjustments.

7.3 Content Authentication

Our system addresses content authentication through one of the following mechanisms:

7.3.1 Summary (Authentication Methods Overview)

- **(1) MOD + Embedded Manifest**: The producer generates all chunks and signs the manifest at once; the consumer first verifies the manifest and then compares the hashes chunk by chunk.
- **(2) Live + Hash Chain**: The producer constructs hash dependencies sequentially for live streaming chunks and periodically signs the chain head with a public key. The consumer verifies the key chunk's signature and checks hash chaining to ensure the integrity of the entire live stream.

7.3.2 Media on Demand + Embedded Manifest

This solution is suitable for on-demand scenarios, utilizing a **one-time public key signature** manifest to batch-verify the integrity of all video chunks.

1.1 Producer Side

1. **Video Chunking and Naming**
    - The video TitleID is split into equal-duration chunks (e.g., every 2 seconds), named as follows:

$$\texttt{/ndn/video/<TitleID>/<Resolution>/chunk=}\langle i \rangle.$$

- Example: `/ndn/video/TitleA/2K/chunk=0,...`

2. **Generating and Signing the Manifest**

   - Compute the hash for each chunk chunk $= 0, 1, \ldots, N-1$:

   $$\text{hash}_i = \text{SHA}-256(\text{Content}_i).$$

   - Write the list $\{(\text{chunkName}_i, \text{hash}_i)\}$ into a manifest object named as:

   `/ndn/video/<TitleID>/<Resolution>/manifest`.

   - Sign the manifest data packet using the producer's private key and store the signature in `SignatureValue`, while specifying the signature algorithm and `KeyLocator` in `SignatureInfo`.

1.2 Consumer Side

1. **Retrieving and Verifying the Manifest**

   - Request `/ndn/video/<TitleID>/<Resolution>/manifest` from the network to obtain the manifest data.
   - Extract the signature information and public key locator (`KeyLocator`), perform public key verification, and if valid, parse all $(\text{chunkName}_i, \text{hash}_i)$.

2. **Chunk Request and Hash Verification**

   - For each listed chunk name chunkName_i, send an Interest to fetch data.
   - Upon receiving the data:

   $$\text{hash}_i' = \text{SHA}-256(\text{Content}_i).$$

   If $\text{hash}_i' = \text{hash}_i$ (matching the manifest), the chunk is valid; otherwise, discard and retry.

7.3.3 Live Streaming + Hash Chain

For frequent live data that requires rapid authentication and scalability, the **hash chain** strategy can be used: signing only "key chunks" with a public key, while verifying the rest via hash chaining to reduce per-chunk signature overhead.

3.1 Producer Side

1. **Continuous Chunk Generation**

   - The live stream is sliced into chunks (e.g., every 2 seconds), named as:

     `/ndn/live/<ChannelID>/<Resolution>/chunk=`$\langle seq \rangle$.

2. **Building Hash Links**

   - For chunk i, calculate:

     $$\text{hash}_i = \text{SHA}-256\big(\text{Content}_i \,\|\, \text{hash}_{i-1}\big),$$

     where hash_{i-1} is the previous chunk's hash.

   - Store hash_i and hash_{i-1} in the chunk's `MetaInfo` fields (e.g., `MetaInfo.thisHash = hash_i`, `MetaInfo.prevHash = hash_i-1`).

3. **Periodic Key Chunk Signing**

   - Every k chunks or time interval Δt, the producer signs the current head hash_i using a private key.

   - The signature is placed in the chunk data (`SignatureValue`) or a separate `signature-block` object.

3.2 Consumer Side

1. **Retrieving Key Chunks**

   - Request key chunks (e.g., `chunk=m`), verify the signature using the producer's public key to confirm the trustworthiness of the chain head hash.

2. **Sequential Hash Chain Verification**

   - For chunks $m+1, m+2, \ldots$, compute:

$$\text{hash}'_i = \text{SHA}{-}256\big(\text{Content}_i \,\|\, \text{hash}_{i-1}\big).$$

     Compare with the declared value in the data. If consistent, the chunk is valid; otherwise, reject.

   - When reaching the next key chunk, verify its signature to ensure the chain's integrity.

7.4 Forwarder Information Exchange Protocol (TO Be Done)

In order to enable distributed decision-making and a global view of network conditions, forwarders must periodically exchange local state information. We are considering leveraging NDN's native Synchronization protocols (e.g., **PSync**) to share this information in near real time. Key design considerations include what information should be shared among forwarders and how to structure the synchronization groups for optimal scalability.

7.5 Baseline model: Distributed variant with local optimization

The model implements a bottom-up optimization strategy where each forwarder makes local decisions based on its downstream users' demands and available bandwidth. These decisions propagate upward, with higher-level forwarders further optimizing within the constraints set by their downstream nodes' decisions.

8 Simulation Results

Runtime testing of Centralized Models We tested Centralized Models in terms of runtime and performance. As shown in Table 3.

Client Assignments and Link Bandwidth Usage Table 4 provides details of client assignments and link bandwidth usage during the simulation. The system assigns resolutions based on the client's bandwidth capabilities and the network's available resources. Link utilizations are dynamically managed to ensure optimal performance.

Figure 2 and 3 illustrates the network topology and link utilizations.

Table 3: Runtime testing of Centralized Models

Users	**Centralized Solve Time (s)**
8	0.004630
16	0.003888
32	0.005997
64	0.011005
128	0.031518

Table 4: Simulation Results: Client Assignments and Link Bandwidth Usage

Client	**Resolution**	**Bandwidth (Mbps)**	**From**	**To**	**Link Usage (Mbps)**
A	1K	8	Server	Core_Fwd_1	269
B	4K	45	Core_Fwd_1	Edge_Fwd_A	69
C	8K	200	Core_Fwd_2	Edge_Fwd_B	200
D	8K	200	Core_Fwd_2	Edge_Fwd_B	200
E	2K	16	Core_Fwd_2	Edge_Fwd_C	24
F	4K	45	Edge_Fwd_C	Client_F	69
G	4K	45	Core_Fwd_2	Edge_Fwd_C	69
H	4K	45	Core_Fwd_1	Edge_Fwd_A	69
I	4K	45	Edge_Fwd_B	Client_I	69
J	4K	45	Edge_Fwd_C	Client_J	69

Next Steps. Next Steps will focus on implementing the model in real-world environments and extending simulations to evaluate the system's robustness under diverse conditions.

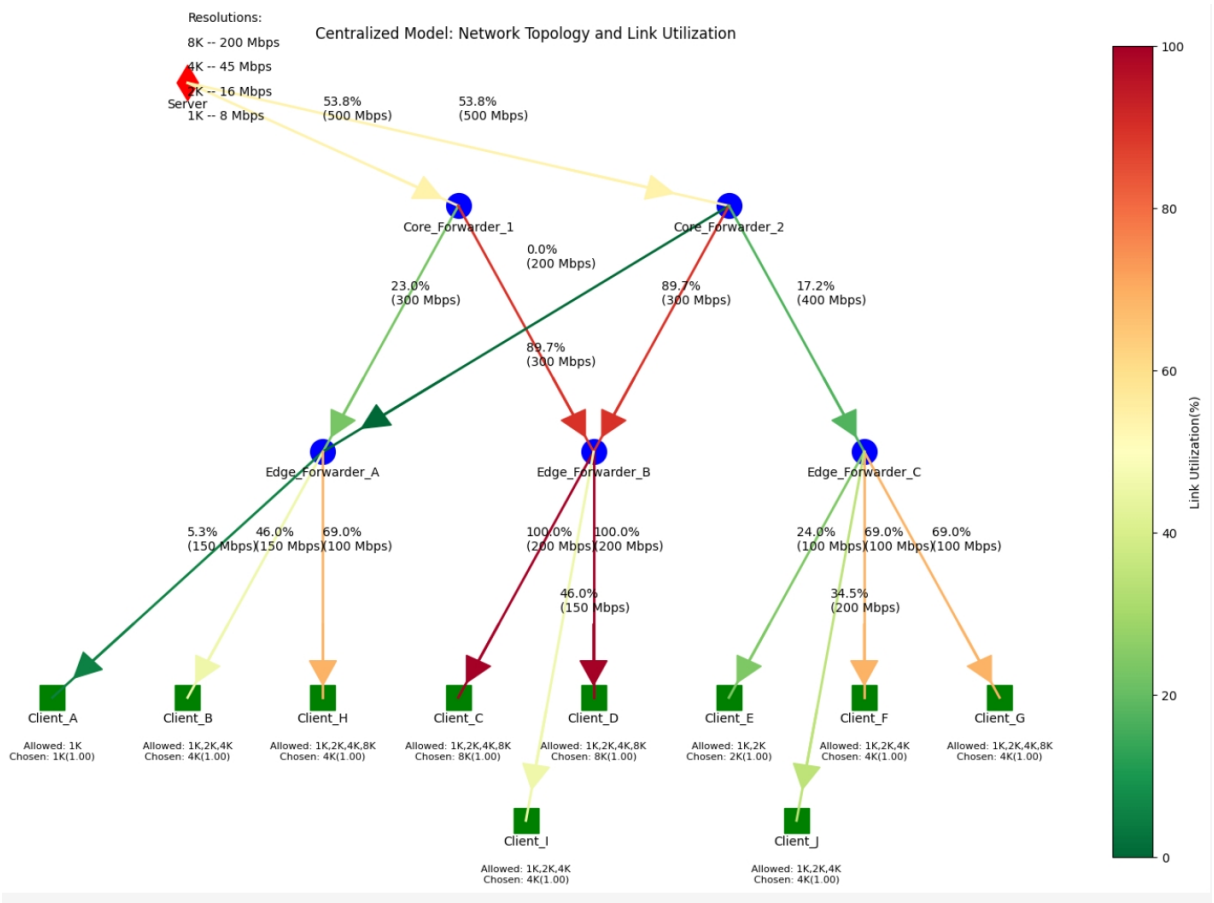

Figure 2: Centralized network topology and link utilization results. The topology includes core and edge forwarders, with varying link utilizations represented by color intensity. Clients are assigned resolutions based on their bandwidth capabilities and network conditions.

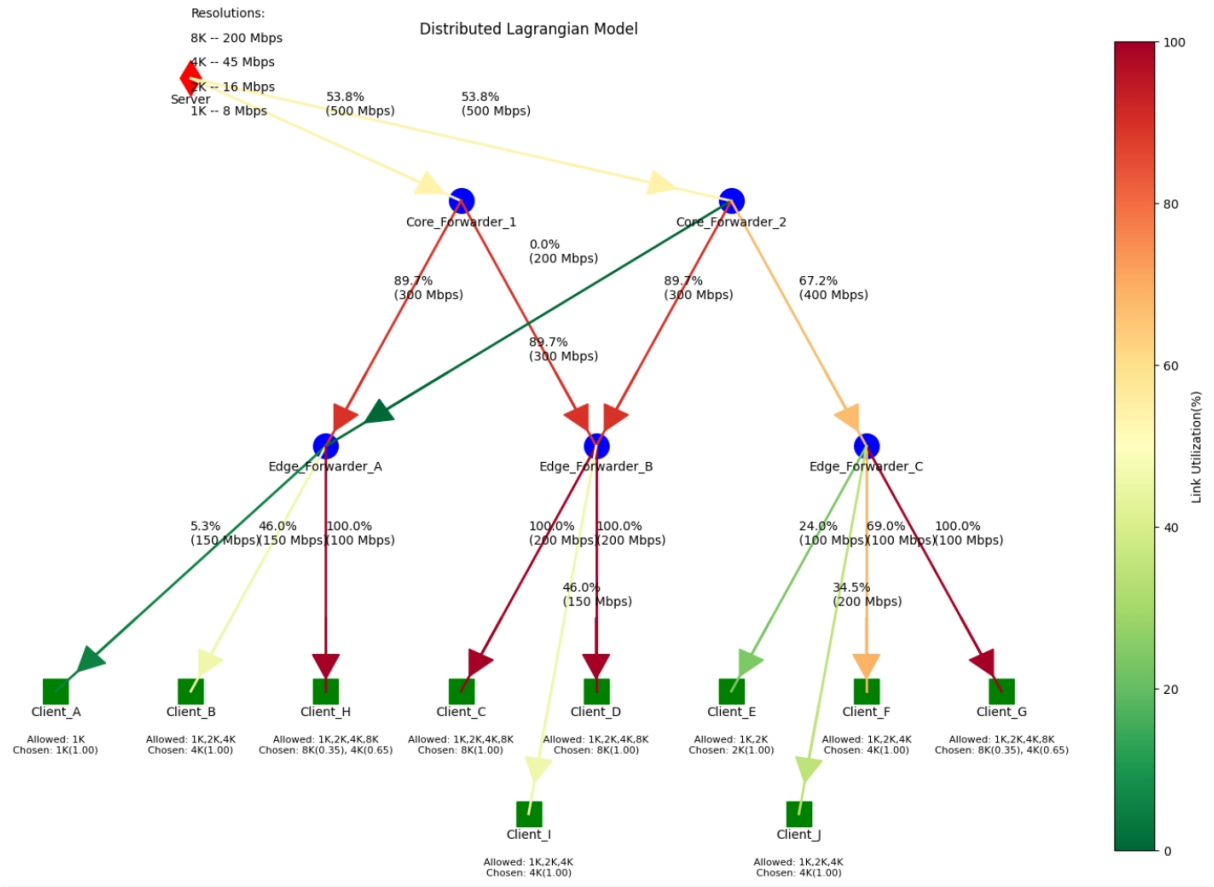

Figure 3: Distributed network topology and link utilization results.

9 Research Plan Overview

The following section outlines the planned research tasks and milestones from February 2025 to February 2027. These tasks build upon the current research progress and aim to extend the proposed system's scalability, adaptability, and applicability to emerging scenarios, such as multi-source multi-destination (MSMD) communication and Metaverse applications. The overarching goal is to establish a unified, efficient, and adaptive multimedia delivery architecture.

9.1 Research Timeline and Milestones

9.1.1 February 2025 – August 2025: Enhancing Quality Adaptation Mechanisms

1. **Dynamic Preference-Based Quality Adaptation:** Extend the current resolution adaptation mechanism to incorporate client preferences. For instance, clients could indicate preferred resolutions (e.g., Client B prefers 4K but accepts 2K with lower priority). The system will dynamically adjust resolutions while considering both network conditions and client preferences. This requires designing new algorithms to balance preference satisfaction and network utilization.

2. **Quality Upgrade Scheme:** Design and implement mechanisms for incremental quality upgrades. Forwarders will periodically probe network conditions (e.g., by sending dummy packets) to determine if additional bandwidth is available. If so, they will trigger downstream nodes to upgrade their resolutions. This approach minimizes client-side complexity and ensures scalability.

3. **Evaluation with Complex Network Topologies:** Test the system in more complex graph structures, including multiple overlapping paths between clients and servers. Develop multipath strategies to exploit modern devices' multi-network connectivity (e.g., WiFi and cellular). Specifically:

    - **Replication Strategy:** Send Interests across all available paths to maximize reliability and resilience against loss.
    - **Best-Path Strategy:** Select the optimal path based on real-time network metrics (e.g., delay, loss rate) and dynamically switch paths when performance degrades.

Expected Deliverables:

- Algorithm for dynamic preference-based quality adaptation.
- Prototype implementation of the quality upgrade mechanism.
- Comparative analysis of replication and best-path strategies in multipath scenarios.

9.1.2 September 2025 – February 2026: Extending to Multi-Source Multi-Destination (MSMD) Communication

1. **Dynamic MSMD Media Distribution:** Expand the system to support MSMD scenarios where multiple producers and consumers interact in real-time, such as online conferences or Metaverse applications. Investigate new mechanisms to handle traffic in different directions and mitigate inter-stream interference.
2. **Resynchronization Mechanisms for Dynamic Participants** Develop mechanisms to resynchronize participants who lose connection or join active sessions. This includes synchronizing producer states with consumer requirements while maintaining QoE.
3. **Traffic Optimization for MSMD Applications:** Design scheduling schemes to minimize the impact of cross-traffic in MSMD scenarios. This involves optimizing data delivery paths to ensure fairness and low latency for all participants.

Expected Deliverables:

- Framework for MSMD communication with real-time synchronization.
- Optimization algorithms for managing cross-traffic and ensuring fairness.

9.1.3 March 2026 – August 2026: Investigating Advanced Media Formats and Representation

1. **Integration of Advanced Media Formats:** Explore the use of new media representations, such as volumetric video and 3D scene data (e.g., USD, glTF). Specifically, investigate the use of hierarchical structures like **octrees** for adaptive data delivery:
    - Represent 3D models as octrees, where each level contains increasing detail.
    - Clients with low bandwidth request top-level nodes, while high-bandwidth clients fetch all nodes.

2. **In-Network Transformation and Transcoding:** Extend the system to support in-network transcoding and transformation for diverse media types. For example, forwarders could dynamically convert 3D scene representations into lower-resolution formats for bandwidth-constrained clients.

Expected Deliverables: - Octree-based data representation and delivery mechanism. - Prototype for in-network transformation and transcoding of volumetric video.

9.1.4 September 2026 – February 2027: Thesis Writing and Defense Preparation

9.2 Key Achievements by February 2027

By the end of the research period, the following milestones will be achieved:

- Development of a fully functional prototype for hierarchical multimedia delivery with preference-based adaptation, multipath strategies, and MSMD support.
- Implementation and evaluation of advanced media formats and in-network transcoding mechanisms.
- Design of a unified media distribution protocol for real-time and interactive applications.
- Comprehensive evaluation of the proposed system in diverse scenarios, including Metaverse applications, dynamic network environments, and multi-user interactions.

References

[1] W. Zhang, S. Lin, F. H. Bijarbooneh, H.-F. Cheng, T. Braud, P. Zhou, L.-H. Lee, and P. Hui, "Edgexar: A 6-dof camera multi-target interaction framework for mar with user-friendly latency compensation," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. EICS, pp. 1–24, 2022.

[2] E. Games, "Unreal networking architecture," https://docs.unrealengine.com/udk/Three/NetworkingOver

[3] W. C. Ng, W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, and C. Miao, "Unified resource allocation framework for the edge intelligence-enabled metaverse," in *ICC 2022-IEEE International conference on Communications*. IEEE, 2022, pp. 5214–5219.

[4] R. Cheng, N. Wu, M. Varvello, S. Chen, and B. Han, "Are we ready for metaverse? a measurement study of social virtual reality platforms," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 504–518.

[5] L. Zhao, Y. Cui, Z. Liu, Y. Zhang, and S. Yang, "Adaptive streaming of 360 videos with perfect, imperfect, and unknown fov viewing probabilities in wireless networks," *IEEE Transactions on Image Processing*, vol. 30, pp. 7744–7759, 2021.

[6] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 197–210.

[7] H. Wang, R. Zhang, C. Li, Z. Xue, Y. Peng, X. Pang, Y. Zhang, S. Ren, and S. Shi, "Twist: A multi-site transmission solution for on-demand video streaming," *Proceedings of the ACM on Networking*, vol. 2, no. CoNEXT2, pp. 1–19, 2024.

[8] L. Giuliano, C. Lenart, and R. Adam, "TreeDN- Tree-based CDNs for Live Streaming to Mass Audiences," Internet Engineering Task Force, Internet-Draft draft-ietf-mops-treedn-07, Aug. 2024, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-mops-treedn/07/

[9] I. Wijnands, E. Rosen, A. Dolganow, T. Przygienda, and S. Aldrin, "Rfc 8279: Multicast using bit index explicit replication (bier)," USA, 2017.

[10] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[11] L. Curley, K. Pugin, S. Nandakumar, and V. Vasiliev, "Warp - Live Media Transport over QUIC," Internet Engineering Task Force, Internet-Draft draft-lcurley-warp-04, Mar. 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-lcurley-warp/04/

[12] K. Pugin, A. Frindell, J. C. Ferret, and J. Weissman, "RUSH - Reliable (unreliable) streaming protocol," Internet Engineering Task Force, Internet-Draft draft-kpugin-rush-02, May 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-kpugin-rush/02/

[13] M. Scharf and A. Ford, "Multipath tcp (mptcp) application interface considerations," Tech. Rep., 2013.

[14] J. Gruessing and S. Dawkins, "Media Over QUIC - Use Cases and Requirements for Media Transport Protocol Design," Internet Engineering Task Force, Internet-Draft draft-gruessing-moq-requirements-05, May 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-gruessing-moq-requirements/05/

[15] M. Papalini, G. Carofiglio, A. Compagno, A. Mantellini, L. Muscariello, J. Samain, and M. Sardara, "On the scalability of webrtc with information-centric networking," in *2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*. IEEE, 2020, pp. 1–6.

[16] G. Carofiglio, G. Grassi, L. Muscariello, M. Papalini, and J. Samain, "Robust: a reliable and flexible media transport for real-time services," *IEEE Transactions on Network and Service Management*, 2023.

[17] M. Elbadry, F. Ye, P. Milder, and Y. Yang, "Pub/sub in the air: A novel data-centric radio supporting robust multicast in edge environments," in *2020 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2020, pp. 257–270.

[18] X. Yu, D. Owens, and D. Khazanchi, "Building socioemotional environments in metaverses for virtual teams in healthcare: A conceptual exploration," in *Health Information Science: First International Conference, HIS 2012, Beijing, China, April 8-10, 2012. Proceedings 1*. Springer, 2012, pp. 4–12.

[19] W. C. Ng, W. Y. B. Lim, J. S. Ng, S. Sawadsitang, Z. Xiong, and D. Niyato, "Optimal stochastic coded computation offloading in unmanned aerial vehicles network," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[20] D. Virgilio, "What comparisons between second life and the metaverse miss," *Online verfügbar unter https://slate. com/technology/2022/02/second-life-metaversefacebook-comparisons. html (abgerufen am 14.07. 2022)*, 2022.

[21] Y. K. Dwivedi, L. Hughes, A. M. Baabdullah, S. Ribeiro-Navarrete, M. Giannakis, M. M. Al-Debei, D. Dennehy, B. Metri, D. Buhalis, C. M. Cheung *et al.*, "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management*, vol. 66, p. 102542, 2022.

[22] M. Xu, W. C. Ng, W. Y. B. Lim, J. Kang, Z. Xiong, D. Niyato, Q. Yang, X. S. Shen, and C. Miao, "A full dive into realizing the edge-enabled metaverse: Visions, enabling technologies, and challenges," *IEEE Communications Surveys & Tutorials*, 2022.

[23] K. Laeeq, "Metaverse: why, how and what," *How and What*, 2022.

[24] B. of Apps, "Pokémon go statistics," https://www.businessofapps.com/data/pokemon-go-statistics/, 2022.

[25] Y. Han, D. Guo, W. Cai, X. Wang, and V. C. Leung, "Virtual machine placement optimization in mobile cloud gaming through qoe-oriented resource competition," *IEEE transactions on cloud computing*, vol. 10, no. 3, pp. 2204–2218, 2020.

[26] B. Games, "Beat saber," *Game [Oculus VR]. Beat Games, Prague, Czech Republic*, 2019.

[27] E. Games, "Fortnite," *Epic Games*, 2017.

[28] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Communications, caching, and computing for mobile virtual reality: Modeling and tradeoff," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7573–7586, 2019.

[29] Y. Cai, J. Llorca, A. M. Tulino, and A. F. Molisch, "Ultra-reliable distributed cloud network control with end-to-end latency constraints," *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2505–2520, 2022.

[30] Q.-V. Pham, X.-Q. Pham, T. T. Nguyen, Z. Han, D.-S. Kim *et al.*, "Artificial intelligence for the metaverse: A survey," *arXiv e-prints*, pp. arXiv–2202, 2022.

[31] Y. Wang and J. Zhao, "A survey of mobile edge computing for the metaverse: Architectures, applications, and challenges," *arXiv preprint arXiv:2212.00481*, 2022.

[32] R. Cheng, N. Wu, S. Chen, and B. Han, "Reality check of metaverse: A first look at commercial social virtual reality platforms," in *2022 IEEE conference on virtual reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2022, pp. 141–148.

[33] S. T. Thomdapu, P. Katiyar, and K. Rajawat, "Dynamic cache management in content delivery networks," *Computer Networks*, vol. 187, p. 107822, 2021.

[34] M. Abbasi, M. Khosravi, and A. Ramezani, "Intelligent resource management at the network edge using content delivery networks," *Enterprise Information Systems*, vol. 17, no. 5, p. 2037159, 2023.

[35] M. Carrascosa and B. Bellalta, "Cloud-gaming: Analysis of google stadia traffic," *Computer Communications*, vol. 188, pp. 99–116, 2022.

[36] W. Zhang, F. Qian, B. Han, and P. Hui, "Deepvista: 16k panoramic cinema on your mobile device," in *Proceedings of the Web Conference 2021*, 2021, pp. 2232–2244.

[37] H. Iqbal, A. Khalid, and M. Shahzad, "Dissecting cloud gaming performance with decaf," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, pp. 1–27, 2021.

[38] C. Holmberg, S. Hakansson, and G. Eriksson, "Web real-time communication use cases and requirements," Tech. Rep., 2015.

[39] S. Shi, V. Gupta, M. Hwang, and R. Jana, "Mobile vr on edge cloud: a latency-driven design," in *Proceedings of the 10th ACM multimedia systems conference*, 2019, pp. 222–231.

[40] N. Pfund, N. Sampat, and J. Viggiano, "Relative impact of key rendering parameters on perceived quality of vr imagery captured by the facebook surround 360 camera," *Frameless*, vol. 1, no. 1, p. 24, 2019.

[41] J. M. P. Van Waveren, "The asynchronous time warp for virtual reality on consumer hardware," in *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, 2016, pp. 37–46.

[42] S. S. A. Shah *et al.*, "Media processing in video conferences for cooperating over the top and operator based networks," Master's thesis, 2012.

[43] I. Aliyu, N. Ko, T.-W. Um, and J. Kim, "A dynamic partial computation offloading for the metaverse in in-network computing," *arXiv preprint arXiv:2306.06022*, 2023.

[44] F. Tang, X. Chen, M. Zhao, and N. Kato, "The roadmap of communication and networking in 6g for the metaverse," *IEEE Wireless Communications*, 2022.

[45] L. Han and K. Smith, "Problem Statement: Transport Support for Augmented and Virtual Reality Applications," Internet Engineering Task Force, Internet-Draft draft-han-iccrg-arvr-transport-problem-01, Mar. 2017, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-han-iccrg-arvr-transport-problem/01/

[46] V. Petrov, M. Gapeyenko, S. Paris, A. Marcano, and K. I. Pedersen, "Standardization of extended reality (xr) over 5g and 5g-advanced 3gpp new radio," *arXiv preprint arXiv:2203.02242*, 2022.

[47] K. Raaen, "Response time in games: Requirements and improvements," *University of Oslo, PhD Thesis*, 2016.

[48] A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi, and G. Fichtinger, "Plus: open-source toolkit for ultrasound-guided intervention systems," *IEEE transactions on biomedical engineering*, vol. 61, no. 10, pp. 2527–2537, 2014.

[49] R. Krishna and A. Rahman, "Media Operations Use Case for an Extended Reality Application on Edge Computing Infrastructure," Internet Engineering Task Force, Internet-Draft draft-ietf-mops-ar-use-case-12, Jul. 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-mops-ar-use-case/12/

[50] G. Van der Auwera, P. T. David, and M. Reisslein, "Traffic and quality characterization of single-layer video streams encoded with the h. 264/mpeg-4 advanced video coding standard and scalable video coding extension," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 698–718, 2008.

[51] B. Juurlink, M. Alvarez-Mesa, C. C. Chi, A. Azevedo, C. Meenderinck, A. Ramirez, B. Juurlink, M. Alvarez-Mesa, C. C. Chi, A. Azevedo *et al.*, "Understanding the application: An overview of the h. 264 standard," *Scalable Parallel Programming Applied to H. 264/AVC Decoding*, pp. 5–15, 2012.

[52] J. Iyengar and M. Thomson, "Rfc 9000 quic: A udp-based multiplexed and secure transport," *Omtermet Emgomeeromg Task Force*, 2021.

[53] D. Brunello, I. Johansson, M. Ozger, and C. Cavdar, "Low latency low loss scalable throughput in 5g networks," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–7.

[54] K. De Schepper, "Rfc 9331: The explicit congestion notification (ecn) protocol for low latency, low loss, and scalable throughput (l4s)," 2023.

[55] S. Jun, Y. Kang, J. Kim, and C. Kim, "Ultra-low-latency services in 5g systems: A perspective from 3gpp standards," *ETRI journal*, vol. 42, no. 5, pp. 721–733, 2020.

[56] M. U. Lokumarambage, V. S. S. Gowrisetty, H. Rezaei, T. Sivalingam, N. Rajatheva, and A. Fernando, "Wireless end-to-end image transmission system using semantic communications," *IEEE Access*, 2023.

[57] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic transmission via revising modules in conventional communications," *IEEE Wireless Communications*, vol. 30, no. 3, pp. 28–34, 2023.

[58] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2020.

[59] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: architectures, applications, and tech-

nical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[60] H. Du, D. Niyato, C. Miao, J. Kang, and D. I. Kim, "Optimal targeted advertising strategy for secure wireless edge metaverse," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 4346–4351.

[61] N. Cheng, H. Jingchao, Y. Zhisheng, Z. Conghao, W. Huaqing, L. Feng, Z. Haibo, and S. Xuemin, "6g service-oriented space-air-ground integrated network: A survey," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 1–18, 2022.

[62] P. K. R. Maddikunta, S. Hakak, M. Alazab, S. Bhattacharya, T. R. Gadekallu, W. Z. Khan, and Q.-V. Pham, "Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17 608–17 619, 2021.

[63] P. Bhattacharya, D. Saraswat, D. Savaliya, S. Sanghavi, A. Verma, V. Sakariya, S. Tanwar, R. Sharma, M. S. Raboaca, and D. L. Manea, "Towards future internet: The metaverse perspective for diverse industrial applications," *Mathematics*, vol. 11, no. 4, p. 941, 2023.

[64] J. Collins, "Gprs tunneling protocol (gtp)," in *Encyclopedia of Cryptography, Security and Privacy*. Springer, 2022, pp. 1–3.

[65] A. Shahmansoori, B. Uguen, G. Destino, G. Seco-Granados, and H. Wymeersch, "Tracking position and orientation through millimeter wave lens mimo in 5g systems," *IEEE Signal Processing Letters*, vol. 26, no. 8, pp. 1222–1226, 2019.

[66] J.-H. Choi and D.-J. Shin, "Generalized rach-less handover for seamless mobility in 5g and beyond mobile networks," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1264–1267, 2019.

[67] F. Y. Okay and S. Ozdemir, "Routing in fog-enabled iot platforms: A survey and an sdn-based solution," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4871–4889, 2018.

[68] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath quic: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.

[69] G. Sinha, M. R. Kanagarathinam, S. R. Jayaseelan, and G. K. Choudhary, "Cquic: Cross-layer quic for next generation mobile networks," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–8.

[70] W. Yang, S. Shu, L. Cai, and J. Pan, "Mm-quic: Mobility-aware multipath quic for satellite networks," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 608–615.

[71] Y. Zhang, Z. Xia, S. Mastorakis, and L. Zhang, "Kite: Producer mobility support in named data networking," in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, 2018, pp. 125–136.

[72] J. Auge, G. Carofiglio, L. Muscariello, and M. Papalini, "Anchorless mobility through hICN," Internet Engineering Task Force, Internet-Draft draft-auge-dmm-hicn-mobility-04, Jul. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-auge-dmm-hicn-mobility/04/

[73] T. Wang, Z. Su, Y. Xia, J. Muppala, and M. Hamdi, "Designing efficient high performance server-centric data center network architecture," *Computer Networks*, vol. 79, pp. 283–296, 2015.

[74] R. Cheng, N. Wu, S. Chen, and B. Han, "Will metaverse be nextg internet? vision, hype, and reality," *IEEE Network*, vol. 36, no. 5, pp. 197–204, 2022.

[75] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user vr 360°: A qoe-aware deep learning-aided multicast framework," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2491–2508, 2020.

[76] R. Brandenburg, R. van Koenen, and D. Sztykman, "Cdn optimization for vr streaming," in *Amsterdam: International Broadcasting Convention*, 2017.

[77] A. Mahzari, A. Taghavi Nasrabadi, A. Samiei, and R. Prakash, "Fov-aware edge caching for adaptive 360 video streaming," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 173–181.

[78] P. Maniotis and N. Thomos, "Viewport-aware deep reinforcement learning approach for 360 video caching," *IEEE Transactions on Multimedia*, vol. 24, pp. 386–399, 2021.

[79] Z. Gurel, T. Erkilic Civelek, A. Bodur, S. Bilgin, D. Yeniceri, and A. C. Begen, "Media over quic: Initial testing, findings and results," in *Proceedings of the 14th Conference on ACM Multimedia Systems*, 2023, pp. 301–306.

[80] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, S. Hammoudi, and R. Zimmermann, "Toward one-second latency: Evolution of live media streaming," *arXiv preprint arXiv:2310.03256*, 2023.

[81] J. Cenzano, "facebookexperimental/webcodecs-capture-play: Live streaming low latency experimentation platform in the browser (using webcodecs)." https://github.com/facebookexperimental/webcodecs-capture-play, April, 2023.

[82] A. Azgin, R. Ravindran, and G. Wang, "Scalable multicast for content delivery in information centric networks," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 105–111.

[83] G. Fioccola, P. Mendes, J. Burke, and D. Kutscher, "Information-Centric Metaverse," Internet Engineering Task Force, Internet-Draft draft-fmbk-icnrg-metaverse-01, Jul. 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-fmbk-icnrg-metaverse/01/

[84] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis," in *Image Analysis: 21st Scandinavian Conference, SCIA 2019, Norrköping, Sweden, June 11–13, 2019, Proceedings 21*. Springer, 2019, pp. 28–40.

[85] A. R. Bharambe, J. Pang, and S. Seshan, "Colyseus: A distributed architecture for online multiplayer games." in *NSDI*, vol. 6, 2006, pp. 12–12.

[86] J. Müller and S. Gorlatch, "Rokkatan: scaling an rts game design to the massively multiplayer realm," *Computers in Entertainment (CIE)*, vol. 4, no. 3, pp. 11–es, 2006.

[87] T. Abdelzaher, M. Caesar, C. Mendis, K. Nahrstedt, M. Srivastava, and M. Yu, "Challenges in metaverse research: An internet of things perspective," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*. IEEE, 2023, pp. 161–170.

[88] A. Basu, J. Acharya, T. Karnik, H. Liu, H. Li, J.-S. Seo, and C. Song, "Low-power, adaptive neuromorphic systems: Recent progress and future directions," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 6–27, 2018.

[89] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE communications surveys & tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[90] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4804–4814, 2018.

[91] A. Erfanian, F. Tashtarian, A. Zabrovskiy, C. Timmerer, and H. Hellwagner, "Oscar: On optimizing resource utilization in live video streaming," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 552–569, 2021.

[92] A. Erfanian, H. Amirpour, F. Tashtarian, C. Timmerer, and H. Hellwagner, "Lwte-live: Light-weight transcoding at the edge for live streaming," in *Proceedings of the Workshop on Design, Deployment, and Evaluation of Network-Assisted Video Streaming*, 2021, pp. 22–28.

[93] C. Ge, N. Wang, W. K. Chai, and H. Hellwagner, "Qoe-assured 4k http live streaming via transient segment holding at mobile edge," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1816–1830, 2018.

[94] D. Kutscher, T. Karkkainen, and J. Ott, "Directions for Computing in the Network," Internet Engineering Task Force, Internet-Draft draft-kutscher-coinrg-dir-02, Jul. 2020, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-kutscher-coinrg-dir/02/

[95] Y. Cai, J. Llorca, A. M. Tulino, and A. F. Molisch, "Joint compute-caching-communication control for online data-intensive service delivery," *IEEE Transactions on Mobile Computing*, 2023.

[96] S. M. Rashid, I. Aliyu, I.-K. Jeong, T.-W. Um, and J. Kim, "Graph neural network for in-network placement of real-time metaverse tasks in next-generation network," *arXiv preprint arXiv:2403.01780*, 2024.

[97] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on mobile computing*, vol. 11, no. 5, pp. 821–834, 2011.

[98] N. Zhao, X. Liu, Y. Chen, S. Zhang, Z. Li, B. Chen, and M.-S. Alouini, "Caching d2d connections in small-cell networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 326–12 338, 2018.

[99] O. Hashash, C. Chaccour, W. Saad, K. Sakaguchi, and T. Yu, "Towards a decentralized metaverse: Synchronized orchestration of digital twins and sub-metaverses," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 1905–1910.

[100] M. Beck and T. Moore, "How we ruined the internet," *arXiv e-prints*, pp. arXiv–2306, 2023.

[101] M.-J. Montpetit, "In Network Computing Enablers for Extended Reality," Internet Engineering Task Force, Internet-Draft draft-montpetit-coin-xr-03, Jul. 2019, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-montpetit-coin-xr/03/

[102] D. Kutscher, J. Burke, G. Fioccola, and P. Mendes, "Statement: The metaverse as an information-centric network," *arXiv preprint arXiv:2309.09147*, 2023.

[103] D. K. Jeff Burke, Lixia Zhang, "Named data microverse," https://named-data.net/microverse/.

[104] O. Hiba, H. Leibowitz, and A. Herzberg, "Quicr: Quic resiliency to bw-dos attacks," *Tech. Rep.*, 2020.