M2SR Projet R&D

Applied research combining fog computing and SDN and the realization of innovative solution SDN-F-FDN in video delivery

Etudiant : Linxiang CONG, Yicheng QIAN

Encadré par : Nazim Agoulmine

Décembre 5, 2024

Table de matière

INTRODUCTION

Modern video streaming applications face significant challenges in network performance, scalability, and latency due to the increasing demand for high-quality content and real-time services. To address these challenges, we propose an innovative framework, the SDN-enabled Federation of Fog Delivery Networks (SDN-F-FDN). This solution is designed to combine the programmability and centralized control of Software-Defined Networking (SDN) with the distributed, low-latency, and scalable nature of Fog Computing.

PROPOSED SOLUTION

We provide a comprehensive study of the state-of-the-art survey on integrating Software-Defined Networking (SDN) and Edge Computing for efficient video streaming systems. Our proposed optimizable aspects are optimized as follows:

SDN-F-FDN

Proposal of the SDN-enabled Federation of Fog Delivery Networks (SDN-F-FDN): A novel architecture, SDN-F-FDN, is introduced, leveraging SDN's programmability and edge computing's proximity to users.

This framework enables hierarchical edge servers to collaboratively manage video content delivery, ensuring high-quality streaming with minimal latency.

Bandwidth Allocation and Transmission Optimization

Bandwidth Allocation and Transmission Optimization: We provide a scheme that incorporates adaptive transmission strategies by assigning hierarchical bandwidth levels (e.g., 1080p: level 4, 720p: level 3) for different video qualities and request types.

This ensures optimized link utilization and enhances user experience through intelligent link decisions controlled by SDN controllers.

Intelligent Controller Functions

Intelligent Controller Functions: We analyzed the role of SDN controllers, focusing on their ability to make dynamic link decisions and support single-direction and bi-directional transmissions based on video quality and user demands.

Integration of Video Information Base (VIB)

We proposed architecture includes a Video Information Base (VIB), which records the location of video content across edge servers. This ensures efficient retrieval of videos from the nearest server, reducing latency and improving delivery performance.

Simulation Framework for Hierarchical Edge Servers

Simulation Framework for Hierarchical Edge Servers: A hierarchical simulation framework is developed, where edge servers are interconnected with varying bandwidth capabilities. The framework evaluates the impact of link decisions and bandwidth allocations on streaming quality and overall system performance.

Future Insights and Validation

Future Insights and Validation: The paper identifies future research directions and discusses experimental validations that will demonstrate the SDN-F-FDN architecture's superiority in scalability, latency reduction, and resource efficiency compared to traditional architectures.

MORE DEEPLY

We propose a model that is divided into three main layers: the application layer, the cloud layer, and the edge layer, which communicate with each other over the Internet. As Figure 1.

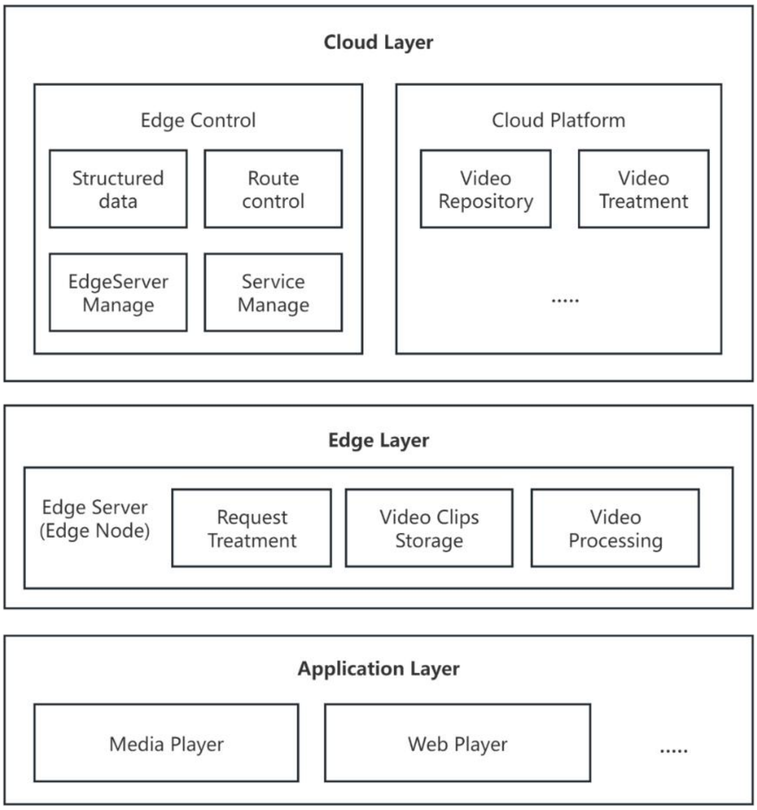

Figure 1: Architecture Conception

The architecture proposed in Figure 1 has 3 layers: Application layer, Cloud layer and Edge layer.

Application Layer: This is the user-facing layer that provides various application services and interactions, e.g., Media Player and Web Player, which are applications that the user touches directly to access content or services.

Cloud layer: This is the data processing and management layer, which is mainly divided into Edge Control and Center Cloud Platform. For Edge Control module, Structured data processing of structured data collected by edge devices, Routing Control manages the communication path from edge devices to the cloud, Edge Server

Management coordinates and monitors the status of edge servers, and Service Management manages the services provided, including service scheduling and resource allocation. And for Cloud Platform, Video Repository stores video content, Video Treatment handles uploaded videos, such as compression and transcoding.

Edge Layer: This is a distributed computing layer close to the user and the device, Edge Server provides low latency computing services to the user, Request Treatment performs initial processing of user requests, Video Clips Storage locally stores user-generated or requested video clips, and Video Processing preprocesses video at the edge (e.g., editing, format conversion).

Core Components and Design

Hierarchical Edge Servers: The architecture employs a hierarchy of edge servers to cache and deliver video content efficiently. By distributing video segments among local fog nodes, the system reduces the reliance on centralized cloud servers and minimizes latency. This caching mechanism ensures that frequently accessed content (hot content) is readily available at the edge, improving quality of experience (QoE).

Video Information Base (VIB): A centralized repository, the VIB, records the location of video segments across the network. It enables the system to dynamically retrieve the nearest available video source, optimizing data delivery paths and reducing latency.

Dynamic Bandwidth Allocation: The framework dynamically allocates bandwidth based on video quality requirements. For example, high resolution videos such as 1080p receive higher-priority links compared to lower-resolution content like 480p. This adaptive mechanism ensures efficient utilization of network resources.

Adaptive Streaming Protocol: The system employs HTTP Live Streaming (HLS) as the primary streaming protocol, enabling adaptive bitrate streaming. HLS divides video into small HTTP-based segments, which can be dynamically adjusted to match network conditions and user device capabilities. This ensures seamless playback even in fluctuating network environments.

Key Features

Low Latency: By caching content at local edge servers and minimizing data transmission from central servers, the system achieves ultra-low-latency streaming.

Scalability: The hierarchical structure allows the framework to scale across regions, supporting a growing user base and increasing content demand.

Resource Optimization: The integration of SDN provides centralized traffic management and resource allocation, ensuring efficient use of available bandwidth and computing power.

Advancement: Compared to traditional CDN and cloud-based solutions, the proposed SDN-F-FDN framework addresses key limitations such as high latency, inefficient resource utilization, and limited scalability. By combining the strengths of SDN and Fog Computing, this framework enables real-time decision-making, adaptive streaming, and enhanced QoE for users in diverse network conditions.

Use case

We take two scenarios as example:

1) when ES(A) has required resources

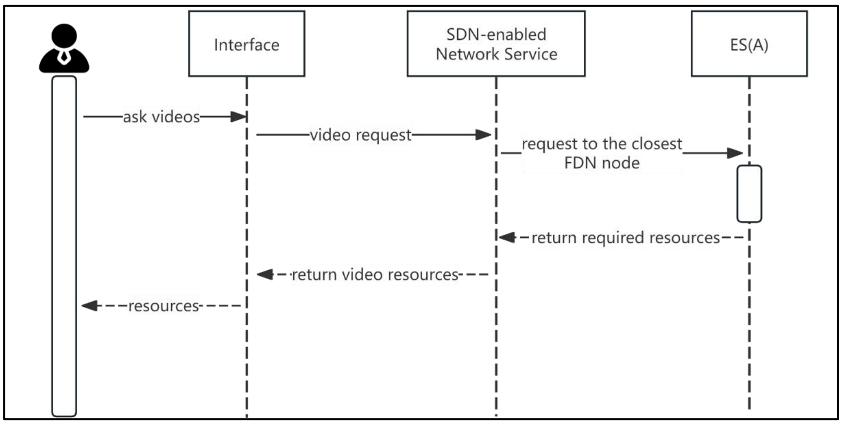

Figure 2: ES(A) has the required resources

Figure 2 show the first use case. The user sends a video request through the Interface, which forwards the request to an SDN-enabled network service. the SDN network receives the request and searches for the edge node (FDN node, e.g., ES(A)) that is closest to the user to process the request, thus reducing network latency. Once a suitable node is found, the SDN network forwards the request to this node, which retrieves the required video resource and returns it to the SDN network, where the resource is then returned to the user through the interface.

2) when E(B) hasn't required resources.

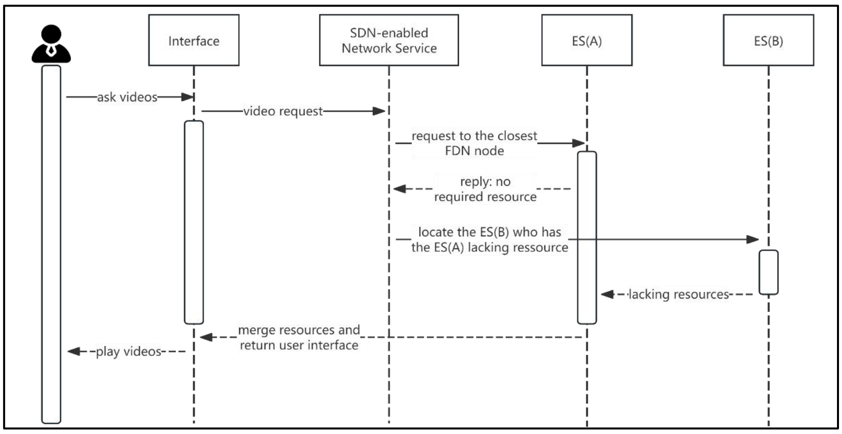

Figure 3: ES(A) hasn't the required resources

Figure 3 shows the second case. The user first sends a video request through the Interface, which forwards the request to an SDN-enabled network service, which sends the request to the nearest edge node (ES(A)). If the ES(A) is unable to fulfill the request (i.e., some resources are missing), it returns a "no resources needed" response. At this point, the SDN service looks for another edge node (ES(B)) that has the missing resource. Once ES(B) is located, the SDN network asks the node to send the missing resources to ES(A), which then combines the complete resources and returns them to the user for playback.

FUTURE WORK

In the near future, our focus will be on implementing a concrete realization of the proposed scheme by leveraging Mininet in combination with SDN controllers such as Ryu, POX, and others to establish the network architecture.

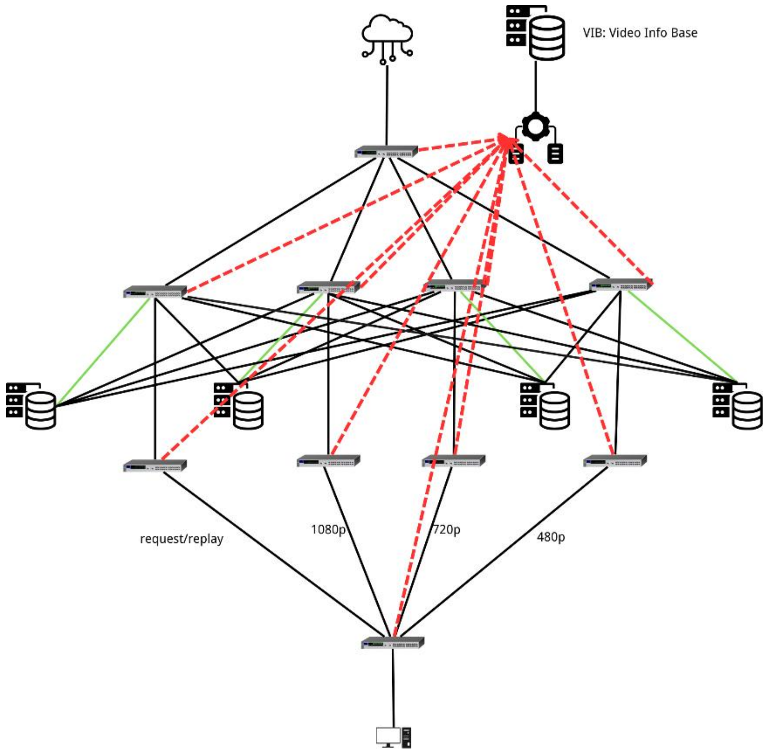

Figure 4: Proposed Network Topology for SDN-F-FDN Implementation

Figure 4 shows the topology that we will build the in December, this process will involve designing interface names and their sequence, assigning corresponding IP and MAC addresses, and configuring link properties such as latency, jitter, and packet loss rate. Using Mininet, we will create the topology and rigorously test its connectivity. Furthermore, we will implement HTTP Live Streaming (HLS) in Python, upload multiple videos of varying quality to the edge servers, and evaluate their initial playback performance. To establish a performance baseline, we will also build a traditional Client/Server (C/S) topology and upload videos for comparative analysis.

In January, we will conduct a comprehensive performance evaluation. This will involve comparing the SDN+Fog model with the traditional C/S model to assess performance improvements in terms of latency, throughput, and other key metrics. These comparisons will validate the advantages of the proposed architecture and offer valuable insights for further optimization and scalability.

CONCLUSION

In this paper, we proposed an innovative framework, SDN-enabled Federation of Fog Delivery Networks (SDN-F-FDN), to address the challenges of video streaming in modern network architectures. The framework integrates the programmability of Software-Defined Networking (SDN) with the distributed nature of Fog Computing, offering a scalable, adaptive, and low-latency solution for high-definition video delivery.

The proposed architecture leverages hierarchical edge servers to cache video content, utilizing the Video Information Base (VIB) for efficient content retrieval. By incorporating HTTP Live Streaming (HLS) as the streaming protocol, the system supports adaptive bitrate streaming, ensuring seamless playback under varying network conditions and user requirements. Different bandwidth levels are dynamically allocated to video streams (e.g., 1080p, 720p, and 480p) and request/replay interactions, optimizing resource utilization and improving user experience.

The SDN controller plays a crucial role in the framework, enabling real-time decision-making for traffic routing and link optimization. High-bandwidth links are prioritized to deliver high-definition video content, while the controller efficiently manages single-direction and bi-directional data flows, further enhancing system performance.

Through its design, the SDN-F-FDN framework addresses the limitations of traditional Client/Server (C/S) and Content Delivery Network (CDN) architectures, reducing latency, enhancing scalability, and improving overall network efficiency. Future work will focus on validating the proposed system through extensive simulations and real-world implementations to further demonstrate its advantages in latency reduction, throughput enhancement, and load balancing across large-scale deployments.

This study provides a foundation for the development of next-generation video streaming systems, paving the way for improved user experiences and efficient resource management in high-bandwidth, low-latency applications.