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

CONG Linxiang
Student, M2CNS
Paris-Saclay University
Evry site
20230886@etud.univ-evry.fr

QIAN Yicheng
Student, M2CNS
Paris-Saclay University
Evry site
20230774@etud.univ-evry.fr

Abstract

Video streaming has become an essential technology in modern life and work, necessitating highly efficient, high-quality delivery systems. Traditional Client/Server (C/S) architectures and early-generation Content Delivery Networks (CDNs), however, are increasingly inadequate for handling the demands of ultra-high-definition (UHD) streaming and the rapid growth of video caching. To address these challenges, we propose the SDN-enabled Federation of Fog Delivery Networks (SDN-F-FDN), an advanced framework that integrates Software-Defined Networking (SDN) with the Federation of Fog Delivery Networks (F-FDN). The SDN-F-FDN extends the core functionalities of F-FDN by introducing programmable and adaptive transmission strategies tailored to varying video quality requirements. By leveraging SDN's dynamic capabilities, the architecture optimizes routing, enhances resource allocation, and balances network load across hierarchical edge and fog computing environments. This significantly reduces latency, improves bandwidth utilization, and minimizes reliance on centralized servers, enabling seamless delivery of ultra-low-latency and high-quality video streams. Future experiments will validate that SDN-F-FDN outperforms traditional architectures in terms of latency reduction, scalability, and network efficiency.

1 INTRODUCTION

The rapid advancement of network technologies has made Software-Defined Networking (SDN), Edge Computing, and Video Streaming integral to modern network architectures. Their integration significantly improves data transmission efficiency and user experience, addressing the demands of high-bandwidth, low-latency applications.

-**Software-Defined Networking (SDN)**
SDN separates the control plane from the data plane, enabling centralized management and dynamic resource allocation. This architecture enhances traffic control, reduces congestion, and supports efficient load balancing, making it ideal for video streaming, where high performance and adaptability are critical.

-**Edge Computing**
Edge computing brings computation and storage closer to users, reducing latency and improving reliability. By caching and processing video content at the network edge, it addresses the limitations of cloud-based solutions, ensuring faster and more reliable streaming services.

-**Video Streaming**
Video streaming dominates internet traffic, driven by the demand for ultra-high-definition (UHD) formats like 4K and 8K, and real-time applications. However, traditional Client/Server (C/S) and Content Delivery Networks (CDNs) are increasingly inadequate, requiring innovative solutions to meet modern demands.

-**SDN and Edge Computing Integration**
Combining SDN's centralized control with edge computing's low-latency capabilities offers a powerful solution for video streaming. Together, they enable intelligent traffic management, efficient resource utilization, and consistent high-quality delivery.

This study explores the integration of SDN and edge computing in video streaming, focusing on their roles in traffic management, load balancing, and resource optimization to establish a framework for next-generation streaming systems. The following table represents comparison of our survey to previously conducted surveys, as table 1

Table 1: Comparison of our survey with previously conducted studies.

Reference	Archi*	Streaming	TM*
[6] 2017	SDN	P2P	
[7] 2019	SDN	DASH	
[8] 2024	SDN+CDN	HAS	FR*
This Paper	SDN+FDN	HLS	SEG*

Archi: Architecture
TM: Transmission Mode
FR: Flexible routing, enabling the delivery of packets within the same network
SEG: Network segmentation, facilitating the delivery of packets across different networks

This paper provides a comprehensive study of the state-of-the-art survey on integrating Software-Defined Networking (SDN) and Edge Computing for efficient video streaming systems. The primary contributions of this work are as follows:
-**Comprehensive Analysis of SDN-Edge Architectures**: The paper surveys existing architectures such as SDN, SDN+CDN, and SDN+FDN, emphasizing their applications in video streaming scenarios and highlighting their advantages in optimizing resource allocation and traffic management.
-**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**: The study 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**: The role of SDN controllers is analyzed, 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)**: The 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**: 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**: 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.

2 STATE-OF-THE-ART

-Methodology
In order to study the state-of-the-art, we mainly choose the paper databases such as IEEE, ScienceDirect, Web Of Science, Google scholar as our background support. After a detailed review of the project title "*Optimization and Monitoring of Video delivery in Hierarchical Edge/Fog Computing using SDN : Video streaming use case*", we came up with the following keywords: **1)** SDN+FOG, **2)** Video streaming.
First, we went to the database of papers mentioned above and searched for papers containing the keywords **SDN+FOG** from 2018 to 2024, and Table 2 lists the number of papers on this subject held by the major paper databases.
Systems combining SDN and FOG computing technologies enable low latency, high bandwidth utilization and powerful dynamic management capabilities, making them particularly attractive in scenarios requiring real-time decision-making, resource optimization and high reliability. Currently, the research directions of the papers about SDN+FOG searched in the paper repository are: smart city[3], Internet of

Table 2: Number of papers related to SDN+FOG stocked in different databases from 2018 to 2024

Database	Number
IEEE	356
ScienceDirect	1449
GoogleScholar	22300
WebOfScience	391

Things[4], Internet of Vehicles, energy[1] and mission scheduling[5] etc. As Table 3, It shows some of the research areas of the paper, and the PAT* shows the techniques used in the paper in addition to SDN+FOG.

Table 3: Currently principal domain of SDN+FOG's applications

Reference	Domain	PAT*
[6] 2024	IOT	6G
[3] 2023	SmartCity	RAFDA
[2] 2021	Energy	Node Cooperation
[9] 2024	HealthCare	Machine Learning
[4] 2020	Surveillance	YOLOv3

PAT: Principal Architecture or Technology, except SDN+FOG

Then we went and searched for relevant papers containing the keyword Video Streaming and we came up with Table 4.

Table 4: Number of papers related to Video Streaming stocked in different databases from 2018 to 2024

Database	Number
IEEE	15372
ScienceDirect	42209
GoogleScholar	17900
WebOfScience	13850

By searching and analyzing the related papers, we found a prominent and frequently mentioned technology in video streaming delivery: **CDN** architecutre. CDNs[12] typically reduce latency by caching content on geographically distributed servers, which is particularly advantageous for video streaming applications. By storing video content closer to end users, CDNs enable faster delivery and reduce buffering times, especially for frequently accessed "hot" content, such as live streams or trending videos. However, this approach has significant limitations. While "hot" content benefits from CDN caching, the majority of "cold" content, such as less popular videos, experiences a low cache hit rate, leading to inefficient use of cache storage. This inefficiency becomes more pronounced in streaming systems with extensive content libraries, where a significant proportion of content may rarely be accessed.Although SDCDN solution [13] has been proposed recently, this does not fundamentally solve the above problems. Furthermore, CDN solutions are often unsuitable for ultra-long-term commitments due to their high deployment costs, geographic constraints, and the variability in service quality across providers. For example, scaling to meet the growing demands of high-definition video streaming or supporting adaptive streaming protocols like HLS and DASH can further amplify these challenges. Although SDCDN solutions [13] have been proposed recently to improve CDN efficiency and scalability, these approaches fail to fundamentally address issues such as dynamic resource allocation, ultra-low latency requirements, and cache space optimization, all of which are critical for modern streaming applications.

While researching the technical architecture of CDNs, we discovered the video streaming transport protocols that most papers will mention: HLS(HTTP Live Streaming) and DASH(Dynamic Adaptive Streaming over HTTP). In video streaming systems, HLS and DASH are two widely adopted adaptive streaming protocols. Table 5 shows the comparison between HLS and DASH on certain metrics.

Table 5: Comparison of DASH and HLS

Metric	DASH	HLS
Compatibility	-	+
Latency	+	+
Standard	+	-
Device Support	+	+
Implementation	-	+
Use Cases	+	+

DASH offers broad support across various platforms but has limited compatibility with Apple devices, whereas HLS provides native support within the Apple ecosystem, making it ideal for users of iOS and macOS. DASH generally achieves lower latency, although HLS with Low-Latency HLS (LL-HLS) offers comparable performance. As an open and widely adopted standard, DASH is accessible and flexible, while HLS is proprietary to Apple, limiting its adoption outside of Apple-centric environments. DASH supports a wide range of devices except within the Apple ecosystem, where HLS excels as the preferred protocol. However, DASH is more complex to implement compared to the simpler architecture of HLS. In terms of use cases, DASH is particularly suitable for low-latency scenarios, while HLS is best suited for applications that prioritize seamless integration within the Apple ecosystem.

Both video streaming protocols are powerful and reliable for online video delivery, offering significant functional equivalence in many aspects. However, their performance characteristics differ in specific scenarios: **1)HLS** is renowned for its superior compatibility, especially with Apple devices such as iPhones, iPads, and MacBooks, which natively support HLS. With over a billion iOS users globally, HLS offers seamless streaming without requiring third-party software. This broad compatibility makes HLS a preferred choice in many applications. **2)DASH**, on the other hand, excels in latency reduction, offering slightly lower delays compared to HLS in live streaming scenarios. However, the latency advantage of DASH over HLS is often negligible in practice, particularly with advancements like Low-Latency HLS (LL-HLS), which bridges the gap.

As Table 3 highlights, the integration of SDN and fog computing has enabled numerous applications across domains like IoT, smart cities, and healthcare. This combination facilitates low latency, dynamic resource management, and high reliability, addressing the limitations of traditional CDN-based solutions. By merging the strengths of adaptive streaming protocols like HLS and DASH with the decentralized and efficient processing capabilities of F-FDN who is based on FOG architecture, next-generation video streaming systems can effectively address the growing demands for low-latency, high-quality content delivery.

HLS and DASH, each of which has its own unique advantages and is suitable for different scenarios. The comparison between HLS and DASH mentioned in Table 5 focuses on compatibility and latency performance: the main advantage of HLS is its broad compatibility, while DASH is characterized by lower latency. However, although DASH has a slight advantage in latency, the improvement over HLS is not significant.

As highlighted in this article, there is a significant equivalence in functionality between HLS and MPEG-DASH. Both protocols are powerful and reliable for online video streaming.[11] However, we believe that compatibility is the factor that tips the balance in favor of HLS. HLS is simply much more widely compatible than MPEG-DASH. There are over one billion iOS users worldwide, which means that most of them, unless using third-party browsers, cannot play MPEG-DASH video streams.[11]

Finally, our eyes are focused on papers that contain both SDN, FOG and Video Streaming, the table below shows the number of papers derived from searching for the above three keys in the major paper databases.

Table 6: Number of papers related to SDN+FOG+Video Streaming stocked in different databases from 2018 to 2024

Database	Number
IEEE	< 15
ScienceDirect	< 728
GoogleScholar	< 11200
WebOfScience	< 9

After a thorough search, we found a paper[7] containing both SDN, Fog and streaming keywords.The paper[7] investigates a dynamic task deployment approach for SDN-based fog and cloud interoperability in stream processing applications, aiming to meet the requirements of low latency and high resource utilization. The authors propose a series of dynamic algorithms (e.g., FogGreedy+ and NAAO) to coordinate fog and cloud resources via SDN controllers to optimize task allocation under limited network bandwidth. however, the reference to stream processing in this paper[7] is a broader concept not limited to video streaming.

After a lot of searching and reviewing, we found a paper that uses only fog and streaming techniques: F-FDN solution [10]. The main feature of F-FDN is that a video can be broken up into chunks and stored on separate FDNs. The based solution F-FDN advocates a Federated Fog Delivery Network (F-FDN) based on fog computing is proposed for low-latency video streaming. F-FDN achieves regionalized caching of video content by integrating distributed fog computing nodes (Fog Delivery Networks (FDNs)) and cloud computing resources and on-demand processing, thus providing high-quality streaming services (QoE) among users in different regions. on-demand processing to provide high-quality streaming services (QoE) among users in different regions.F-FDN can cache popular video segments locally and share cached content among neighboring FDNs, while transcoding and processing video through edge nodes, reducing the dependence on central cloud servers.

Streaming latency is primarily influenced by two key factors [8]: the time required for video segment processing and the duration of network transmission. The F-FDN architecture relies on the quality of network connectivity between fog nodes and between nodes and cloud servers. When network or link conditions are poor, system performance may be severely affected, including increased latency and decreased throughput.

Therefore, on the original solution, the introduction of SDN technology and the dedicated network architecture can be a good solution to the problem of over-reliance on link quality in the Fog architecture, and SDN is able to select high-quality transmission links

3 PROPOSED SOLUTION

To address the challenges associated with video streaming in modern network architectures, we propose an innovative framework called the SDN-enabled Federation of Fog Delivery Networks (SDN-F-FDN) whcich is base on F-FDN[10]. This framework integrates the programmability and centralized control of Software-Defined Networking (SDN) with the distributed and low-latency nature of Fog Computing. It aims to offer a scalable, adaptive, and efficient solution for high-quality video streaming.

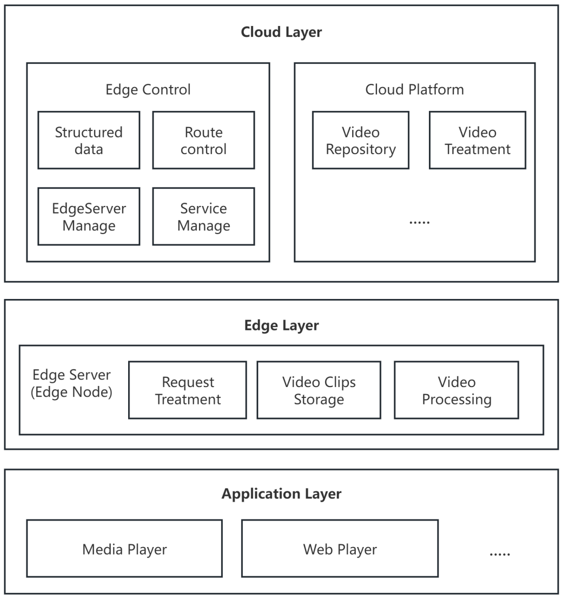

Figure 1: Architecture Conception

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.

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.

Topology

Figure 2 shows the topology that we build, this process involves 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 created the topology and rigorously tested its connectivity. Furthermore, we implemented 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 also test the traditional Client/Server (C/S) topology performance and upload videos for comparative analysis.

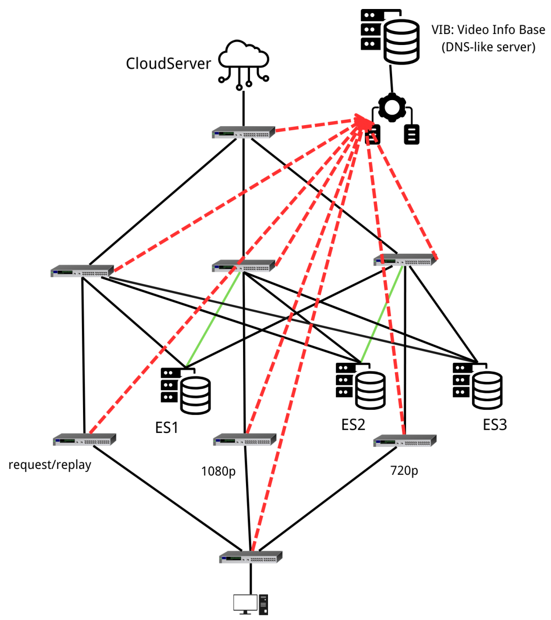

Figure 2: Proposed Network Topology for SDN-F-FDN Implementation

Sequence Diagram

Users access the video service through the

HLS.html page, which includes a built-in RESTful client. Using an HTTP POST request, users can specify the video they want to access. A location server runs a RESTful service on the backend, receiving the user request and calculating the optimal Edge Server's IP address using an algorithm. It then checks whether the requested video resource is available on the Edge Server.
We take two scenarios as example: 1) when ES has required resources, 2) when ES hasn't required resources. Figure 3 shows the first sequence diagram. The client sends a server ip request through the Interface, which forwards the request to location server, it processes the request and return the ip of the closest edge server. After the client receives the ip, it sends a video request to the edge server who has this ip. Once the edge server receives the request, it return the required resource to the client.

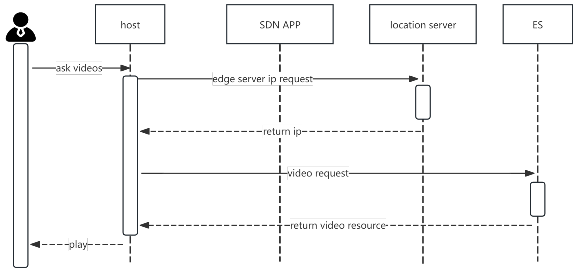

Figure 3: ES has the required resources

Figure 4 shows the second case: when ES hasn't required resources. The user first sends server ip request through the Interface, which forwards the request to an location server, it processes the request and find that the closest edge server does not have the required resource. So it returns the cloud server ip. Once the client receives the cloud server ip, it sends a video request. When the ryu controller detects that the dst ip is cloud server. It is not normal, so it finds the closest server who is closed to the client and sends a Notification message who is based on UDP protocol to the closest server. This Notification message contains the information about the missing chunks' names. When the closest server receives the message, it gets the missing chunks from the cloud server based on the information in the message.

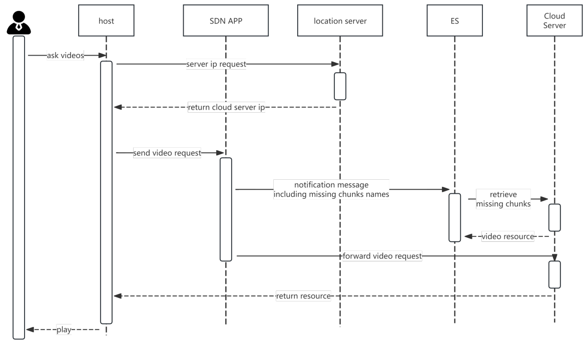

Figure 4: ES hasn't the required resources

4 TEST

We tested the above two scenarios in one physic machine win11. We have 8 VM: 1 host, 1 cloud-server, 3 edge server, 1 location server, 1 ryu controller and 1 VM who runs Mininet topology. in case 1, the average reception time of HLS video player: 5.32 seconds for 1080p video and 4.42 seconds for 720p video. In case 2, that is, requesting video from the cloud server while the edge server pulls resources, this process is more bandwidth-consuming, and the average time for the HLS video player: 22.07 seconds for 1080p video and 20.14 seconds for 720p video.

5 FUTURE WORK

We have also designed a feature that allows the Edge Server's IP address to be transmitted via UDP packets, replacing TCP-based video streaming protocol, in order to realize a more extensive SDN network.

5.1 Effective communication

By using UDP packets to transmit data, this solution is lighter and faster than HTTP-based interfaces, making it suitable for scenarios requiring low latency and real-time communications.

5.2 Strengthening the SDN approach

By transferring more network control logic to the SDN controller, this solution enhances centralized management and intelligent network orchestration.

5.3 High extensibility

This architecture can easily be extended to larger networks, supporting various traffic optimization strategies such as dynamic load balancing or nearest server access.

6 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.

References

[1] Adnan Akhunzada, Sherali Zeadally, and Saif ul Islam. *Power and Performance Efficient SDN-Enabled Fog Architecture.* 2021. DOI: 10.48550/ARXIV.2105.14607. URL: https://arxiv.org/abs/2105.14607.

[2] Adnan Akhunzada, Sherali Zeadally, and Saif ul Islam. “Power and Performance Efficient SDN-Enabled Fog Architecture”. In: *IT Professional* 23.6 (Nov. 2021), pp. 24–30. ISSN: 1941-045X. DOI: 10.1109/mitp.2021.3085840. URL: http://dx.doi.org/10.1109/MITP.2021.3085840.

[3] Muhammad Ibrar et al. “Reliability-Aware Flow Distribution Algorithm in SDN-Enabled Fog Computing for Smart Cities”. In: *IEEE Transactions on Vehicular Technology* 72.1 (Jan. 2023), pp. 573–588. ISSN: 1939-9359. DOI: 10.1109/tvt.2022.3202195. URL: http://dx.doi.org/10.1109/TVT.2022.3202195.

[4] S. Pan and Y. Yuan. “System and application of video surrveillance based on edge computing”. In: *telecom science* 36.6 (2020), pp. 64–69.

[5] Linh-An Phan et al. “Dynamic fog-to-fog offloading in SDN-based fog computing systems”. In: *Future Generation Computer Systems* 117 (Apr. 2021), pp. 486–497. ISSN: 0167-739X. DOI: 10.1016/j.future.2020.12.021. URL: http://dx.doi.org/10.1016/j.future.2020.12.021.

[6] Sonia Ben Rejeb and Nidal Nasser. “Optimizing Video Surveillance in IoT 6G with Edge/Fog Computing”. In: *ICC 2024 - IEEE International Conference on Communications.* IEEE, June 2024, pp. 4792–4798. DOI: 10.1109/icc51166.2024.10622190. URL: http://dx.doi.org/10.1109/ICC51166.2024.10622190.

[7] Michał Rzepka et al. “SDN-based fog and cloud interplay for stream processing”. In: *Future Generation Computer Systems* 131 (June 2022), pp. 1–17. ISSN: 0167-739X. DOI: 10.1016/j.future.2022.01.

006. URL: http://dx.doi.org/10.1016/j.future.2022.01.006.

[8] Arman Shojaeifard et al. “Full-Duplex Cloud Radio Access Network: Stochastic Design and Analysis”. In: *IEEE Transactions on Wireless Communications* 17.11 (Nov. 2018), pp. 7190–7207. ISSN: 1558-2248. DOI: 10.1109/twc.2018.2865771. URL: http://dx.doi.org/10.1109/TWC.2018.2865771.

[9] Subhranshu Sekhar Tripathy et al. “An SDN-enabled fog computing framework for wban applications in the healthcare sector”. In: *Internet of Things* 26 (July 2024), p. 101150. ISSN: 2542-6605. DOI: 10.1016/j.iot.2024.101150. URL: http://dx.doi.org/10.1016/j.iot.2024.101150.

[10] Vaughan Veillon, Chavit Denninnart, and Mohsen Amini Salehi. “F-FDN: Federation of Fog Computing Systems for Low Latency Video Streaming”. In: *2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC)*. IEEE, May 2019, pp. 1–9. DOI: 10.1109/cfec.2019.8733154. URL: http://dx.doi.org/10.1109/CFEC.2019.8733154.

[11] Max Wilbert. *HLS vs. MPEG-DASH : comparaison des protocoles de diffusion en direct en 2023 — dacast.com*. https://www.dacast.com/fr/le-blog-des-experts-video/mpeg-dash-vs-hls-ce-quil-faut-savoir/. [Accessed 29-11-2024].

[12] Zhouyun Wu et al. “CDN Convergence Based on Multi-access Edge Computing”. In: *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, Oct. 2018, pp. 1–5. DOI: 10.1109/wcsp.2018.8555887. URL: http://dx.doi.org/10.1109/WCSP.2018.8555887.

[13] Huixiang Yang, Hanlin Pan, and Lin Ma. “A Review on Software Defined Content Delivery Network: A Novel Combination of CDN and SDN”. In: *IEEE Access* 11 (2023), pp. 43822–43843. ISSN: 2169-3536. DOI: 10.1109/access.2023.3267737. URL: http://dx.doi.org/10.1109/ACCESS.2023.3267737.