Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 631–636
DOI: 10.15439/2024F1683
ISSN 2300-5963 ACSIS, Vol. 39

Agent at the Edge: Opportunity and Challenges of Video Streaming Analytics at the CDN Edge

Reza Shokri Kalan
Istinye University, Istanbul-Türkiye
reza.shokri@hotmail.com, reza.kalan@istinye.edu.tr

Seren Gul
Digiturk beIN Media Group, Türkiye
seren.gul@digiturk.com.tr

***Abstract*—To provide high-quality streaming services to end users, streaming analytics applications need to process massive volumes of data promptly. Such applications suffer from high network transmission costs for transferring logs to a stream processor (cloud or on-premises), archiving, and computing costs for timely log analysis due to the volume, variety, and velocity of log data. This is especially important in live streaming, where millions of users play video simultaneously and consume network resources that are technically limited. A Distributed log analytic system can help to deal with this huge amount of data located at different locations and change rapidly. The advent of rich resources at the edge has enabled data processing close to the data source in a geo-distributed setup. Pushing log analytics closer to data sources is an effective strategy to reduce resource bottlenecks for the stream processor. This paper explores the benefits and drawbacks of deploying agents to analyze distributed logs, aiming to enhance the quality of video playback. Where increasing network and client diversity at the edge adds complexity to the task of processing live streams to end users situated across various networks and geographic locations. Furthermore, it introduces a mechanism to provide an abstract overview of the current streaming ecosystem resulting in better QoE.**

***Index Terms*—Adaptive Video Streaming, Log Analytics, Multi-agent, QoE**

I. Introduction

TODAY'S explosively growing Internet video traffic and users' ever-increasing quality of experience (QoE) demand for high-quality video streaming brings tremendous pressure to the Over-the-Top (OTT) providers in the competitive entertainment market. The OTT or content providers promote the Content Delivery Network (CDN) capacity to distribute media content to end users worldwide. A CDN is a group of distributed and interconnected servers that enhance the delivery time of content to end users. As a new efficient network paradigm, CDN edge provides a promising alternative pushing video content closer to the network edge, thus reducing both content access latency and redundant network traffic. However, our long-term tracking analysis shows that different CDNs and networks have variable performance over time. Even the best CDN may have poor quality of service at a particular time or region. CDN's agnostic nature of video streaming makes it possible to switch between alternative CDNs to achieve optimal performance in terms of video quality and service cost. To this end, we need to log analytic and fast reactions. However, users' distribution and networks' highly dynamic traffic patterns make log analysis a difficult task because of big data properties (volume, velocity, and variety).

It is worth emphasizing that, cloud-based big data analytics and decision-making cannot meet the requirements of many latency-sensitive applications [1] such as low-latency live streaming. The traditional log analytics model requires moving collected logs to a central location on the network, such as a data center or cloud. However, many emerging and real-time application use cases require edge analysis capabilities. Pushing the log analytics closer to the data source is an effective strategy to reduce resource bottlenecks (network bandwidth) for the stream processor. This method gives more agility and decreases response time. The result of queries at the edge is combined at a central point in the cloud or data centers.

Leveraging a distributed log analytics system can forward a user's query to any CDN that has multiple edges distributed over the Internet. Thus, it retrieves more relevant results for each query than centralized general-purpose search engines, which operate only within limited data sets. Furthermore, this is a low-cost search solution that can be deployed even on a single computer, because it does not require a large amount of storage and processing power. Furthermore, there is no need to transmit a large volume of data logs to the central analytics system in the cloud. In a multi-agent system, agents can share the same goal and work cooperatively with their neighbors or they can focus on their own goals [2].

Efficient decision-making is a target of intelligent multi-agent systems, where multiple agents communicate and collaborate to solve complex tasks by overcoming individual limitations. To achieve this, a broker mediates between users and different data sources to collect and combine search results. To accomplish this task it is required that each network area has its' own private broker and an extra broker installed in the main domain. A local agent on each endpoint (origin server, network, and edge server) deals with collected raw data and sends query results to the query agent. This paper discusses improving video streaming Quality of experience (QoE) by considering distributed edge analytics capabilities along with core network capacity.

The remainder of the article is structured as follows: background and related works are discussed in Section II. Central log analytics is introduced in Section III. Applied methodology and system architecture are discussed in Section IV. Experi-

mental results are discussed in Section V. We conclude the paper and draw future paths in Section VI.

II. Background and Related Works

A. Background

Adaptive bitrate (ABR) provides clients with optimal video display quality by dynamically adjusting to the appropriate bitrate in real-time. To achieve this goal, i) on the client side, it takes into account the available bandwidth of the network and the capabilities of the client device. ii) On the service provider side, adaptive streaming involves creating multiple copies of video content and distributing it through a CDN network. Fig 1 shows an abstract view of the HTTP Adaptive Streaming (HAS) technology, where a video is encoded in several different bitrates (each bitrate has a different quality) and fed to the origin media server. Each video file is divided into small chunks (e.g., 2 or 4 segments of video). Manifest files hold video metadata including the number of bitrates (or display), the number of video segments, segment size, subtitles, and audio information.

Clients connect to the CDN at the edge and initiate video streaming scenarios by requesting and downloading the manifest file from the nearest edge. The client adjusts the appropriate bit rate according to the download speed and manifest file information. The network has a dynamic nature, when network traffic patterns change, the client switches between different video display qualities (or bitrate) to avoid buffering. When a client requests a video segment, the CDN checks the availability of the requested segment at the edge and returns a fast response if it has already been requested by another client and is cached at the edge. Otherwise, the CDN forwards the client request to the origin server. The origin server implements Just-in-Time packaging, creates the request segment in the appropriate format (DASH, HLS, MSS), and sends it to the CDN [3]. The CDN delivers the video segments to the client while caching them at the edge to be served on subsequent incoming requests. Fig 2 shows the process of Dynamic Adaptive Streaming over HTTP (DASH), which is is the popular format and the only standard form of adaptive streaming technology adopted by many OTTs and vendors.

B. Related Works

The traditional and widely used method is to collect all data sets in a central site before running the query. However, waiting for such centralized aggregation significantly delays the timeliness of the analytics. Minimizing query response times in a geo-distributed setting is critical for live video streaming analytics. Therefore, to overcome this limitation, the natural alternative is to execute the queries close to the data source. There are some research efforts to reduce query delay with more attention to onsite analytics. The bandwidth limitation is considered in [4]. This study uses an online heuristic to redistribute data sets among the sites before the queries' arrival and places the tasks to reduce network bottlenecks during the query's execution. However, in a large-scale network, it is not possible to assume that the network snapshot at any point. Furthermore, the proposed approach is not optimal.

The advent of rich resources at the edge has enabled data processing close to the data source in a geo-distributed setup, thus alleviating the network and compute bottlenecks at a central stream process or running in the cloud which is discussed in [5]. The authors investigated resource availability and resource bottleneck issues that affect query partitioning strategies while streaming analytics. A common form of data analytics in geo-distributed networking is hub and spoke architecture, where spokes run analytics at the edge and send results to the central warehouses or hub. Presented algorithms in [6] aims to address the question of how much computation should be performed at the edges versus the center. The developed algorithm optimizes two key metrics: WAN traffic and staleness (delay in getting results). Authors in [7] present a multi-cloud architecture to evaluate and optimize quality of service (QoS) between end users and multi-cloud CDN operators. According to the evaluation results, the proposed method performs a long-term minimum resource deployment to meet users' requests with higher QoS and lower cost. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics introduced in *JCAB* [8]. The proposed scheme aims to optimize the balance between analytical accuracy, service delay and energy consumption. All those studies focused on network traffic optimization rather than quality Key Performance Indicators (KPI) which are critical for end-user experience quality.

Rather than discussing how to collect and analyze live reports and make decisions based on real-time report analysis, the paper in [9] focuses on deep learning models based on report data. Most of the existing video analytics solutions [10], [11] developed with a focus on resource utilization rather than human-perceived quality optimization. Unlike the quality experienced by users, video analysis algorithms can tolerate dropped frames and poor image quality, which are very important from a QoE perspective.

III. Central Log Analytics System

Log analytics is the activity of obtaining information relevant to research cases from a collection of data resources. Searches can be based on metadata or full-text indexing. As shown in Fig 3, in the central processing logic, all data and processes are aggregated in central On-premise or cloud that can provide enough resources including bandwidth, storage, and processing power.

- *Collection*: Typically, streaming video benefits from multi-CDN architecture, reports from different sources may have different formats for date, time, etc. Therefore, it is necessary to categorize and refine the raw data before analyzing it.

- *Processing*: Unlike individual use cases, monitoring systems consisting of multiple CDNs, each with multiple edges and servers, require powerful processing systems

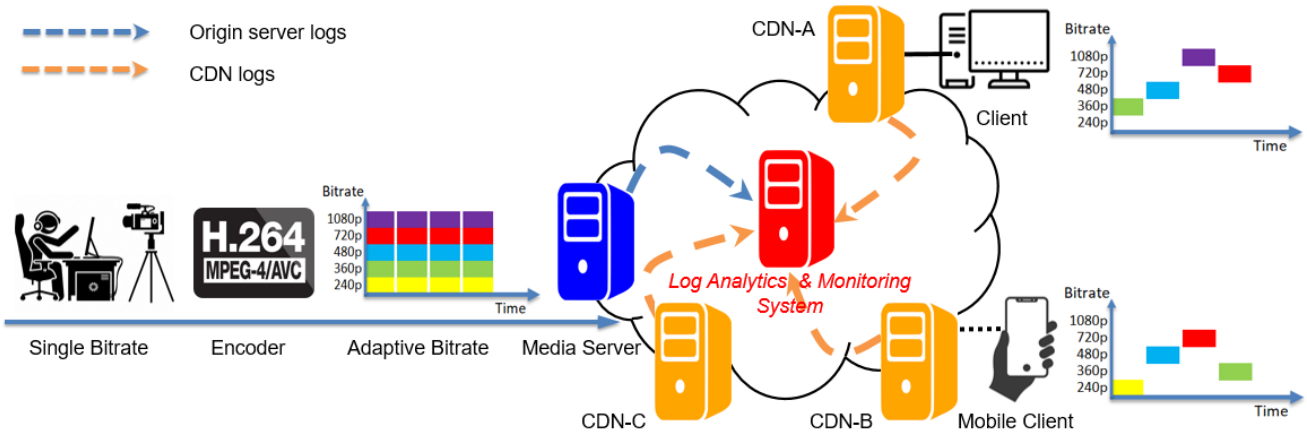

Fig. 1: High overview of adaptive video streaming technology and log analytic system. Clients connect to the CDN-edge and download video segments while dynamically switching to suitable video representations (bitrates).

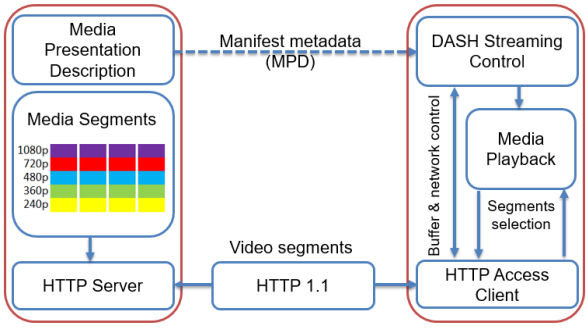

Fig. 2: Dynamic Adaptive Streaming over HTTP (DASH)

to operate in a distributed manner.

- *Wisdom*: To achieve high performance, service providers monitor the system or enter a query in the form of a keyword through the user interface. Finally, analyzes the information retrieved from related sources and then takes appropriate actions as fast as possible.

When live streaming services are a concern, a central processing log analytics system has difficulty meeting big data processing requirements. Different data sources in video streaming include:

- *Origin Media Servers*: Origin media servers store different quality of the same video file each segmented into small chunks, which helps in faster delivery and better adaptation on the client side. Manifest files include metadata information. The objective of live streaming is to provide video services without delay and compromise quality, where hundreds of thousands of online users concurrently connect to the system and display video.
- *Content Delivery Networks*: CDN platforms deploy edge servers to deliver content to end-users or process data close to where it is generated, enabling the identification of bandwidth and network latency issues and real-time response to improve delivery service performance. CDN has information related to related Internet Service providers (ISPs) as well as traces clients' actions and gathers information connected to the CDN from the closest edge.

- *Clients and Media Players*: At the client side, the media player runs an adaptation algorithm and adapts to a suitable bitrate according to the network throughput of buffer occupancy [12]. Being aware of client types (e.g., mobile, smart TV, PC, etc), connection networks (e.g., WiFi, Cellular), and QoE metrics (e.g., average received bitrate, delay, rebuffering) helps to provide better service.

IV. Agent at the Edge of Log Analytics System

Compared with the centralized analytics model, where no processing is performed at the edges, a distributed analytic solution overcomes the limitation of sending all the data to a dedicated centralized location. In geographically distributed streaming analytics domains, (e.g., Google Analytics, Akamai Media Analytics, etc.) data sources send data streams to nearby CDN edges. CDN edge processes incoming data and sends results to a central location where incoming results are summarised, stored, and can be visualized by analytics tools. However, those solutions focus more on network performance and give more priority to QoS rather than QoE. Agents at the edge analytic solution could be an alternative for distributed analytics where agents are created (or destroyed) on-demand near the data source. The proposed system is constructed with three types of agents:

- *Query Agents (QA)*: This type of agent gets queries from users and forwards them to a Resource Manager Agent

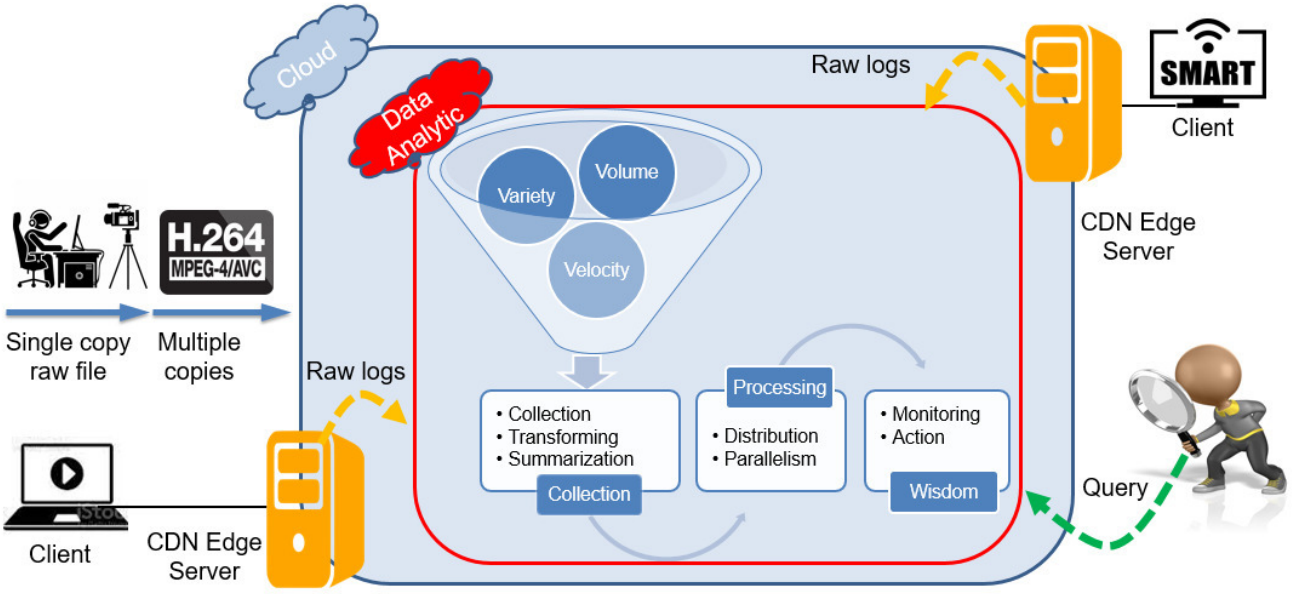

Fig. 3: Integration of cloud and ELK stack for log analytics. Sending raw logs increases time complexity due to the volume of data. Therefore, this architecture may not be suitable for real-time services such as live video streaming.

(RMA). In addition, it gets the results from the Local Agents (LA), merges, and ranks the results.

- *Resource Manager Agent*: It forwards queries to all LAs and manages the information resources. It is responsible for creating and removing LAs for the data resources.
- *Local Agent*: Each resource has its LA. This agent realizes the query processing and sends the result to QAs.

As shown in Fig. 4, the RMA is responsible for managing resources and agents. When a new resource comes or an existing one has been deleted, it creates or deletes the related agent. When a new request comes from a QA, the RMA encapsulates the request and source address inside the new request before forwarding it as a query to LA. The LA processes the query and forwards the result to the QA as it has information about its address. Finally, the QA merges all the results and ranks them for visualization. *Algorithm* 1 illustrates the system activity. According to the algorithm, if the incoming order is a query, then the log analytics function will be active (Fig. 5), otherwise add/remove action flowed.

There is two types of queries; analytic queries and action queries. The resource manager is an agent responsible for managing incoming queries. If a new query is an analytic query, it sends the query to a local agents and waits for the response. According to the query results, if there is no result returned, it displays 'Result not found'. Otherwise, it merges, ranks, and displays the results in a suitable format. The resource manager is also responsible for creating or removing local agents for data resources. When an incoming query is an action query (to add or remove an agent), the resource manager creates or deletes the agent and updates the resource management information. With local agents positioned at CDN-edge points and the RMA centrally located in the cloud with substantial computing resources, this architecture offers enhanced flexibility and efficiency. The query agent forwards analytic queries to local agents at the edge points with the assistance of the RMA and summarizes the incoming results. This approach reduces time complexity by minimizing data transfers, as local agents transmit only final results rather than raw data logs.

Algorithm 1: *System Activity* Algorithm

```
Input: list of incoming request, Query/ Management
Output: update data set/ manage resources
while true do
    Receive incoming request
    if request is a 'query' then
        Distribute query over agent
        if result <0 then
            Print result not find
        else
            Run merge/rank & display results
        end
    else
        if request is a 'add' then
            Create new resource
        else
            Remove resource
        end
        Update resource manager
    end
end
```

V. Experimental Results

To enhance service quality, QoS is considered, which is usually evaluated by delay factors, packet loss rate, or throughput. However, the evaluation of users is more based on the perceived quality. Therefore, OTT is more consistent with QoE parameters to ensure a certain level of video quality for all customers. Even the best CDN has poor quality during the day due to network dynamics. OTT providers use multi-CDN architecture to overcome this limitation. Having an overview of customer distribution and network resources helps CDN service providers choose the best infrastructure for service delivery and error recovery.

The performance of services at the edge depends on the availability of resources, which may often be limited. Typically CDNs have a rich endpoint, however, the high complexity of the edge ecosystem brings additional complexity to log analytic management. For example, each CDN has a specific log strategy restricting access to the data set. Typically, they provide a data stream that technically transports huge amounts of raw logs to central processing in the cloud. This log strategy has three main difficulties: i) Transferring huge amounts of data in a short time needs more bandwidth, furthermore, even the best CDN needs time to gather this data set and forward it to a central processing unit in the cloud. ii) Processing this log in the cloud needs more time and resources. iii) it is hard to save those logs in storage for a long period because of limited capacity. We found that during important live events (e.g. a football game) the size of raw logs can increase to 4GB per minute. To overcome those challenges we consider two different log strategies.

A. Selective parameters analytical model

In live streaming, we have two groups of channels, those with normal traffic and the group with peak traffic. The analytical model of selective logs is used for groups that have normal traffic but consist of more channels. The next group includes channels (for example. sports channels) that are small in number and have certain traffic in normal mode, but at certain times they bring a lot of traffic to the network, meaning more network and processing resources. For normal channels

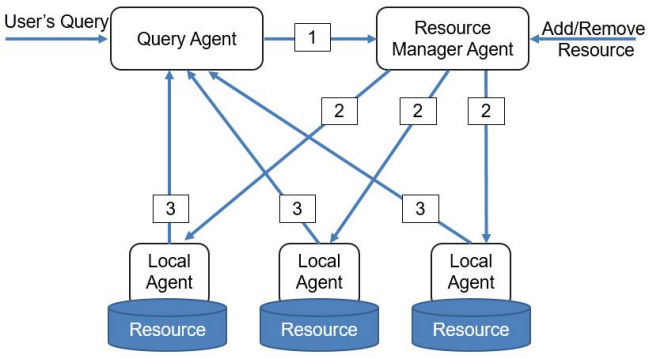

Fig. 4: Overview of proposed system and agents interaction

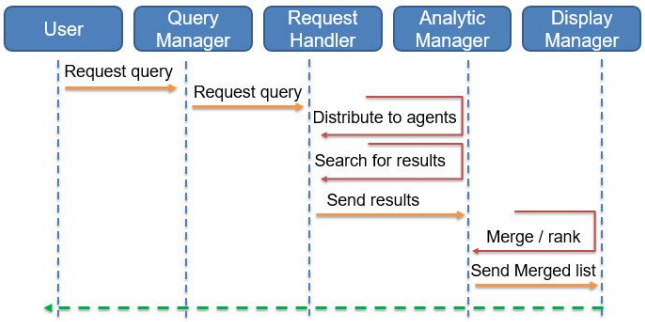

Fig. 5: Sequence diagram of proposed system

(live and VoD), we considered a selective parameter analytic model, where limited parameters are considered for analysis which include but are not limited to HTTP status code, client IP address, response time, etc. This strategy results in fewer raw logs, thus enabling low delivery latency, fast processing, and less storage capacity. As a result, the cost is low.

Table I, compares conventional and selective parameters analytical models for 24 hours, where the selective model provides less volume log and leads to fast transfer and processing time without compromising the quality of analytic results. Technically, different configurations result in degrees of effectiveness, which additionally depends on the application. The selection of appropriate parameters relies on our existing experiment outcome of monitoring live and VoD streaming over the years. Recall, predictive analytics requires historical data and machine learning algorithms to predict future flow trends, optimize resource allocation, and improve overall streaming performance.

B. Selective log analytical model

On the other hand, channels with peak traffic provide live video services to hundreds of thousands of users. CDNs (as well as OTT) prefer to have normal traffic instead of peak traffic because of the difficulty in resource management and traffic distribution. In the selective log analysis model, instead of requesting all raw data logs from the CDN, requesting a portion (sampling rate) of the log results in a faster response from the CDN. As a result, the analysis of smaller amounts of data provides a near-real-time overview of live traffic distribution and users' QoE, enabling online reactions.

Considering the traffic patterns and the number and distribution of online clients in a wide geography we need more parameters to provide seamless traffic. Therefore, an analytic model based on selective parameters is not more applicable for hot events video streaming where end users request and expect higher video quality while connecting via heterogeneous clients (devices) and networks. Furthermore, in terms of security and fraud detection analysing more parameters is essential. In addition, the sampling method provides a random log that can not carry sufficient information. Even though log delivery is fast, it requires more query processing time.

TABLE I: Selective parameters analytical & storage efficiency

Log analytic models	#Parameters	#Received requests	Total size
Conventional	35	171 M/day	73 GB
Selective Parameters	21	171 M/day	44 GB

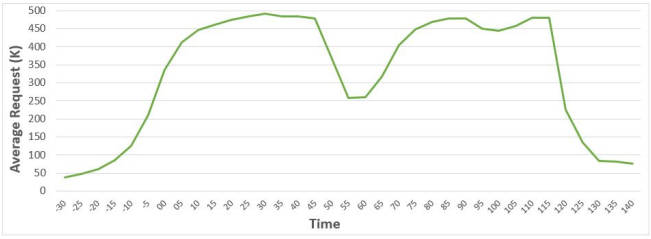

Fig. 6: Abstract overview of received requests at the edges

C. Layoff log analytical model

While technically it is impossible to run agents at the edge points due to CDN policy, we have to gather all raw logs in our central processing unit at the cloud. Although this scenario is more costly regarding processing and storage, it's crucial for log analysis in more detail. Concerning reducing query processing time we move to an alternative "layoff" model that takes a snapshot of all necessary logs in specific intervals and stores results in a separate database. Real-time monitoring outcomes of these logs give an abstract overview of network throughput, available bandwidth, HTTP status code, number of requests, etc. For example, Fig. 6 illustrates an abstract overview of the number of incoming request/s in one football event. During 120 minutes of event streaming, 3.4 billion requests hit the edge, making it difficult to analyze the data in detail and make real-time decisions. However, instance data logs at short intervals (e.g., 5 minutes) give an overview of the video stream and enable quick reactions. Tracing key metrics like bitrate, quality oscillation, and buffering time ensure smooth playback and user perceived QoE.

VI. Conclusion

Even the best CDN has poor quality in the daytime due to network dynamism. To ensure end-user perceived quality, the OTT requires continuous monitoring and analysing of service delivery. This includes ongoing surveillance of origin servers, end-to-end delivery networks, and clients' KPIs to maintain satisfactory end-user satisfaction. Gathering and processing all streaming logs in the central point cloud takes longer and is not feasible for live streaming monitoring. Additionally, due to CDN policy, creating on-demand agents at the CDN edge, where a local agent receives and executes queries on a local dataset and then sends the results to a central administrator where the results are aggregated and visualized, is not applicable. To address those limitations we introduced and implemented "layoff" log analytic method which provides an abstract overview of critical KPI values for streaming.

Gaining comprehensive insight into the origin server, network traffic patterns, allocated bandwidth, and clients' QoE facilitates prompt responsiveness and enables the attainment of optimal streaming performance. Achieving such insights can be facilitated by leveraging AI technologies. In future work, we aim to employ AI to develop models that analyze vast amounts of data generated by the origin server, network infrastructure, and client interactions. These models can detect patterns, predict network congestion or performance issues, and recommend optimizations in real-time to enhance streaming performance. Additionally, AI-powered systems can continuously learn from new data, thereby improving their accuracy and effectiveness over time. AI-assisted real-time monitoring provides immediate insights into the streaming process as it happens, enabling rapid detection and response to issues such as buffering, latency, or fraud detection

Acknowledgment

This research has been supported by Digiturk beIN Media Group (https://digiturk.com.tr), in close cooperation with the R&D team and Istinye University (https://www.istinye.edu.tr/).

References

[1] C. Yang, S. Lan, L. Wang, W. Shen, and G. G. Huang, "Big data driven edge-cloud collaboration architecture for cloud manufacturing: a software defined perspective," *IEEE access*, vol. 8, 2020.

[2] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Applied Intelligence*, vol. 53, no. 11, 2023.

[3] R. S. Kalan, M. Sayit, and A. C. Begen, "Implementation of sand architecture using sdn," in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2018, pp. 1–6.

[4] Q. Pu, G. Ananthanarayanan, P. Bodik, S. Kandula, A. Akella, P. Bahl, and I. Stoica, "Low latency geo-distributed data analytics," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, 2015.

[5] A. Sandur, C. Park, S. Volos, G. Agha, and M. Jeon, "Streaming analytics with adaptive near-data processing," in *Companion Proceedings of the Web Conference 2022*, 2022, pp. 563–566.

[6] B. Heintz, A. Chandra, and R. K. Sitaraman, "Optimizing timeliness and cost in geo-distributed streaming analytics," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 232–245, 2017.

[7] C. Wang, Z. Lu, Z. Wu, J. Wu, and S. Huang, "Optimizing multi-cloud cdn deployment and scheduling strategies using big data analysis," in *2017 IEEE (SCC)*. IEEE, 2017, pp. 273–280.

[8] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 257–266.

[9] R. Bhardwaj, Z. Xia, G. Ananthanarayanan, J. Jiang, Y. Shu, N. Karianakis, K. Hsieh, P. Bahl, and I. Stoica, "Ekya: Continuous learning of video analytics models on edge compute servers," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 119–135.

[10] M. Zhang, F. Wang, and J. Liu, "Casva: Configuration-adaptive streaming for live video analytics," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 2168–2177.

[11] Y. Wang, W. Wang, D. Liu, X. Jin, J. Jiang, and K. Chen, "Enabling edge-cloud video analytics for robotics applications," *IEEE Transactions on Cloud Computing*, 2022.

[12] R. S. Kalan, "Improving quality of http adaptive streaming with server and network-assisted dash," in *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 2021, pp. 244–248.