# Video Sessions KPIs clustering framework in CDNs

Sepideh Malektaji*, Diala Naboulsi*, Roch Glitho*, Alexander Polyantsev†,
Ali El Essaili‡, Cyril Iskander†, Richard Brunner†

*CIISE, Concordia University, Montreal, Quebec, Canada – {s_malekt, d_naboul}@encs.concordia.ca, glitho@ciise.concordia.ca
† Ericsson, Montreal, Quebec, Canada –{alexander.polyantsev, cyril.iskander, richard.brunner}@ericsson.com
‡ Ericsson Research, Herzogenrath, Germany – {ali.el.essaili}@ericsson.com

***Abstract*— Users' viewing experience in the video delivery process is of paramount importance for Content Delivery Networks (CDNs). Throughout their operations, CDN providers target the satisfaction of users' expectations in terms of Quality of Experience (QoE). In this context, CDN providers need to acquire knowledge on users' QoE and correlate observations through different video sessions in order to identify QoE degradations and investigate their potential root cause. In the absence of users' feedback on their QoE, CDN providers can monitor and analyze Key Performance Indicators (KPIs) throughout video sessions. This allows to assess the Quality of Service (QoS) offered to users, influencing their QoE. However, due to the large number of sessions handled by CDN operators, it is not possible to conduct such an analysis manually. In this work, we introduce a framework that allows to automatically group a large set of video sessions into a small number of representative clusters, with each cluster containing video sessions with similar patterns of KPIs. The framework builds upon a set of features representing the evolution of KPIs over a session. It relies on an unsupervised machine learning algorithm to form the clusters. We evaluate the framework over a real-world dataset with traffic logs relating to thousands of sessions. The obtained results underline the capabilities of the proposed framework.**

***Keywords—CDN, KPIs, QoS, Video Traffic Dataset***

## I. INTRODUCTION

Content Delivery Networks (CDNs) consist of surrogate servers that replicate content from an origin server. Surrogate servers are placed in strategic locations to enable an efficient delivery of content to users [1]. CDNs play a vital role in the delivery of content across the Internet today. This is particularly the case for video content, with video traffic forming more than 70% of all consumer internet traffic today [2]. By 2020, the percentage of video traffic is further expected to grow to 82% [2]. CDNs will need to cope with the corresponding growth, straining their infrastructure. In addition, they need to satisfy users' expectations of their viewing experience. There, the rapid expansion of high resolution videos is making users more demanding than ever, putting additional pressure on CDNs.

In this context, it is critical to track and analyze users' perceived viewing experience, to detect corresponding drops and investigate their potential root causes. Such an analysis can be completed based on feedback from users on their viewing experience, through a set of Quality of Experience (QoE) measures. However, this feedback is seldom available. In the absence of such a feedback, CDNs need to rely on Key Performance Indicators (KPIs). KPIs can be computed from network traces collected over the CDN infrastructure. They allow CDN operators to evaluate the Quality of Service (QoS) offered to users, influencing their QoE. For instance, as shown in [3], the video Download Bit Rate (DBR), i.e. the rate at which bits are transferred from the surrogate server to the user, and the video Quality Level (QL), i.e. the bit rate at which the video is encoded, are two main KPIs in adaptive bit rate streaming, that help in determining users' QoE.

The analysis of KPIs' evolution throughout video sessions and the correlation of observations among video sessions allow for investigations into the potential root cause for common drops in QoS and QoE. However, this process is not an easy task. Tens of thousands of video requests are received by a country-wide CDN provider on a daily basis, according to our investigations. Analyzing and correlating the KPIs among corresponding sessions is simply not possible manually. Automated approaches are thus needed to allow for this analysis over a massive set of sessions.

In this work, we focus on the analysis of the evolution of KPIs across video sessions, for QoS and QoE analysis, using unsupervised machine learning tools. We propose a framework that allows the automatic formation of clusters of video sessions, presenting similar evolution of KPIs. We capture the dynamics of KPIs over each session through a set of representative features. Using $k$-means clustering algorithm, we build upon collected features to form clusters of video sessions, with each containing similar sessions in terms of KPIs evolution. The framework is evaluated over a real-world traffic dataset covering thousands of sessions collected over the infrastructure of a country-wide CDN provider. We show that our framework allows to distinguish meaningful clusters.

The rest of the paper is organized as follows. In Sec. II, we present motivating scenarios and derive requirements for a QoS and QoE analysis framework. Sec. III reviews the related work on QoS and QoE analysis in CDNs. In Sec. IV, we introduce the proposed framework. We then cover the evaluation of the framework in Sec. V. Finally, we conclude the work in Sec. VI.

## II. MOTIVATING SCENARIOS AND REQUIREMENTS

Various business actors are involved in the video content delivery process over traditional CDNs: ***i)*** end-users that consume the content; ***ii)*** content providers (e.g. YouTube) that offer the content; ***iii)*** CDN providers (e.g. Akamai) that offer the content delivery service through surrogate servers; and ***iv)*** an Internet Service Provider (ISP) if surrogate servers are deployed in data centers owned by the ISP. Each of these actors can perform actions leading to degradations in the QoS and QoE. We describe two such scenarios that allow us to derive the

978-1-5386-5553-5/19/$31.00 ©2019 IEEE

requirements for a QoS and QoE analysis framework for CDN providers.

The first scenario is a load balancing strategy among CDN surrogate servers, presented in Fig. 1(a). Two servers A and B are shown. Server B has a lower load than server A. As highlighted in the figure, due to the load balancing strategy, a set of users, initially served by server A, are served instead by server B. Such a strategy can lead to more balanced loads among the servers. However, server B is farther from the set of highlighted users than server A. Thus, these users may experience higher latency, leading to a lower DBR and possibly a lower QL, for adaptive bitrate streaming. In turn, this would imply a lower QoS and a lower QoE, which can be identified through the analysis of DBR and QL KPIs patterns over time.

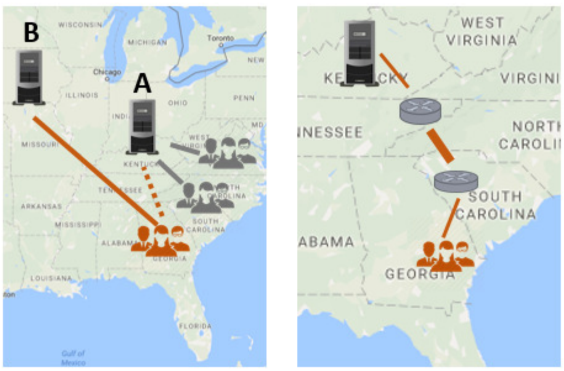
The second scenario is a congestion scenario over a link between two ISP routers. As traffic grows over a communication link, it can get congested. The scenario is presented in Fig. 1(b) with a congested link between two ISP routers, utilized to connect clients to the surrogate server. Congestion can imply the need for retransmissions that lead to fluctuations in the DBR and in turn fluctuations in the video QL for clients, with adaptive bitrate streaming. This can imply drops in QoS and QoE for these clients that can be identified through the analysis of DBR and QL KPIs patterns over time.

For both scenarios, it is critical for the CDN provider to track and analyze KPIs. This would allow the CDN provider to learn about QoS and QoE offered to users, to identify possible drops and to know whether they are caused by issues inside the CDN's domain, e.g. the first scenario, or outside the CDN's domain, e.g. the second scenario. Considering these scenarios, we derive the following requirements for a framework targeting QoS and QoE analysis for CDN providers.

**1. Scalability:** A CDN provider operates over a large scale, handling demand from a large set of users as well as a large set of content. A QoS and QoE analysis framework should therefore operate efficiently over a large scale.

**2. Flexibility:** Multiple KPIs that reflect users' QoE can be collected by the CDN provider. A QoS and QoE analysis framework needs to be flexible to account for multiple KPIs.

**3. Automation:** As analyzing and correlating KPIs cannot be handled manually over a large scale, a QoS and QoE analysis framework needs to offer automated procedures.




(a) Load balancing (b) Link congestion

Fig. 1 Motivating scenarios

**4. Fine granularity:** Users' QoE can fluctuate over time throughout a single video session. A QoS and QoE analysis framework therefore needs to target the analysis of corresponding fluctuations.

## III. Related Work

Many studies have targeted the analysis of QoS and QoE based on CDN traffic datasets. In the following, we present these works and evaluate them according to our requirements.

Several studies have aimed at investigating the impact of specific events in the CDN system on the QoS and QoE over individual video sessions. Fan *et al.* [4] studied the impact of changes in CDN redirection mechanism on the latency perceived by users. The study was conducted over a large scale. However, only one KPI was considered, i.e. latency. Casas *et al.* [5] also analyzed the impact of events in CDNs on the QoE of users. To this end, the authors propose automated approaches that operate over a large scale, and consider multiple KPIs. However, their work does not consider the fluctuations of KPIs throughout sessions. Shafiq *et al.* [6] studied the impact of network dynamics on user abandonment behavior. Their analysis was conducted over a large scale, over multiple KPIs and relies on an automated approach. Nevertheless, KPIs fluctuations over individual sessions were not considered.

Other studies have aimed at identifying the mapping between QoS and QoE metrics in CDNs using traffic datasets. Li *et al.* [7] studied the correlations between video download throughput and user engagement. A large-scale analysis of a single KPI, i.e. download throughput, is conducted. However, it does not account for its fluctuations. Li *et al.* [8] studied correlations between performance and QoE metrics. The correlations were considered for multiple KPIs through automated approaches. However, the evolution of KPIs is not considered, and the analysis is led over a small scale. Orsolic *et al.* [9] predict the QoE level of a session according to a set of KPI features. The evolution of multiple KPIs is considered over individual sessions with an automated approach for prediction. However, the study is conducted over a small scale.

Multiple studies focus on the analysis of anomalies, based on CDN traffic datasets. Giordano *et al.* [10] propose a method to identify changes in CDN cache selection policy. Their method operates over a large scale, considers multiple KPIs and is automated. Nevertheless, it does not account for the evolution of KPIs throughout sessions. Wu *et al.* [11] focus on the detection of video freeze events in video sessions. They propose an automated method that operates over a large scale, considering the evolution of only one KPI over each session, i.e. the inter-segment duration. Dimopoulos *et al.* [12] propose a framework to diagnose the root cause of mobile video QoE issues. The framework adopts an automated approach that covers multiple KPIs. However, it operates over a small scale and does not account for the evolution of KPIs across video sessions. In turn, Zhu *et al.* [13] also target the diagnosis of QoE issues. They propose an automated approach that allows to identify the root cause of large latency increases, over a large scale. Nevertheless, they only account for latency and do not consider its evolution through video sessions.

In summary, none of the previous studies meets all requirements. Some studies were conducted over a small scale, and thus did not meet our scalability requirement. Many works do not consider multiple KPIs, leaving the flexibility requirement unsatisfied. All but two papers introduce automated procedures for their studies, meeting the automation requirement. As for fine granularity, only two studies fulfilled that requirement. While none of the previous works meets all our requirements, our KPIs analysis framework does accomplish that. Our framework forms clusters of video sessions, presenting a similar evolution of KPIs, using unsupervised machine learning tools. To the best of our knowledge, we are the first to investigate the evolution of KPIs throughout video sessions, using unsupervised machine learning tools. Our framework provides a clear understanding of the evolution of KPIs throughout video sessions. It operates by considering the fine-grained evolution of multiple KPIs, throughout each session, it adopts an automated approach and operates over a large scale, thereby meeting all requirements.

## IV. KPIs Clustering Framework

In this section, we present our video sessions clustering framework. Our framework allows video sessions to be grouped into a set of clusters, each associated with a specific pattern in the evolution of KPIs. The evolution of a KPI over time, throughout a video session can be captured through an irregular time series representation. Various clustering approaches can be employed accordingly: raw data-based clustering, feature-based clustering and model-based clustering [14]. Video sessions typically last for a long duration. Therefore, the time series representing the evolution of their KPIs are also long. Moreover, they have unequal lengths, representing user's watching time of a particular video content. In contrast to other approaches, a feature-based clustering approach allows us to cope with these two aspects. We therefore adopt this type of approach in our framework. In the following, we first describe the features that we employ to represent the evolution of KPIs throughout a session. The similarity measure on which the clustering step relies to group sessions is then presented, followed by a description of the clustering algorithm. Finally, we describe how clusters are selected.

### *1- KPIs* Representation

Our framework operates over a set of sessions $S$. We employ $s \in S$ to refer to an individual session in $S$. A session $s$ spans over a time interval $T_s$, a set of time instants $t$. We consider a set $I$ of KPIs. The evolution of KPI $i \in I$ throughout session $s \in S$ is captured via an irregular time series $i_s = \{ i_s^t , \ \forall\, t \in T_s\}$ where $i_s^t$ is the value of KPI $i \in I$ at time instant $t \in T_s$ over session $s \in S$.

We rely on the derived time series representation of KPIs $i_s$ to extract a set of features $V$ that capture the evolution of KPIs throughout a session. The set of features $V$ has been chosen as the set of features considered in [15]. There, the authors showed that the set of features $V$ allows to successfully represent the evolution of a time series. Each feature $v \in V$ captures a specific facet of the evolution of the KPI in question throughout a session. For each KPI $i \in I$, we extract the following six features $V$ for a session $s \in S$ , as detailed below.

**Average.** The average value of KPI $i \in I$ of all samples during session $s \in S$ is obtained with Equation (1).

$$\mu_{s,i} = \frac{1}{|T_s|} \sum_{t \in T_s} i_s^t \tag{1}$$

**Standard deviation.** The standard deviation of KPI $i \in I$, over all its samples in session $s \in S$, is derived with Equation (2).

$$\sigma_{s,i} = \sqrt{\frac{1}{|T_s|} \sum_{t \in T_s} \left(i_s^t - \mu_{s,i}\right)^2} \tag{2}$$

**Skewness**. Skewness is a measure that characterizes the shape of a mathematical distribution. It captures the level of symmetry in the distribution with respect to its central point. For a KPI $i \in I$ over a session $s \in S$, it is derived with Equation (3).

$$W_{s,i} = \frac{1}{|T_s| \sigma_{s,i}^3} \sum_{t \in T_s} (i_s^t - \mu_{s,i})^3 \tag{3}$$

**Kurtosis.** Kurtosis is another measure that allows to characterize the shape of a mathematical distribution. It captures whether the distribution is heavy-tailed or light-tailed. A distribution with a light tail tends to have a low kurtosis. For a KPI $i \in I$ over session $s \in S$, the Kurtosis can be obtained using Equation (4).

$$K_{s,i} = \frac{1}{|T_s| \sigma_{s,i}^4} \sum_{t \in T_s} \left(i_s^t - \mu_{s,i}\right)^4 \tag{4}$$

**Energy.** Energy measures the strength of a time series. It is obtained based on the non-uniform Discrete Fourier transform (DFT) of a time series. Assuming $f_{s,i}^j \in F$ is the $j$th discrete Fourier component for KPI $i \in I$ in session $s \in S$, and that $F$ is the complete set, the Energy is derived with Equation (5).

$$E_{s,i} = \frac{1}{|F|} \sum_{f_{s,i}^j \in F} |f_{s,i}^j| \tag{5}$$

**MLE.** The Maximum Lyapunov Exponent measures the randomness for a time series by quantifying the average logarithmic rate of separation of two nearby subsets of the time series. For a KPI $i \in I$ over a session $s \in S$ it is obtained based on Equation (6).

$$M_{s,i} = \lim_{t \to \infty} \frac{1}{T_s} \, ln \frac{|i_s^{t+\delta} - i_s^t|}{|i_s^{t_0+\delta} - i_s^{t_0}|} \tag{6}$$

### 2- *Similarity Measure*

After deriving the six features for each KPI $i \in I$, over session $s \in S$ , we construct a vector $V_s$ encompassing these features for a session $s \in S$ as follows:

$$V_s = \left[\mu_{s,i}, \sigma_{s,i}, W_{s,i}, K_{s,i}, E_{s,i}, M_{s,i} \;\middle|\; \forall i \in I\right]$$

Each of the features is rescaled to the interval [0,1] by considering the minimum and maximum value of the feature of interest, across all observations. Rescaling is done so that each feature contributes approximately proportionately to the similarity measure. We use $v_{s,i}$ to refer to the rescaled feature $v \in V$ of KPI $i \in I$ over session $s \in S$. This allows us to define a rescaled vector $V_s'$ of features for a session $s \in S$, as follows:

$$V_s' = [\, v_{s,i} \mid \forall i \in I, v \in V]$$

We capture the degree of similarity between a pair of sessions $s$ and $r$ by calculating the Euclidean distance between the corresponding rescaled vectors using Equation (7).

$$d(s,r) = \left[\sum_{v\in V}\sum_{i\in I}\left(v_{s,i} - v_{r,i}\right)^2\right]^{1/2} \quad (7)$$

#### 3- Clustering algorithm

To obtain the set of video sessions clusters, we employ the widely-known, $k$-means clustering algorithm [16]. It has been selected due to its superior performance compared to other approaches, e.g. hierarchical clustering techniques, as shown in the literature [16] as well as our experiments. $k$-means clustering algorithm is an unsupervised machine learning algorithm that allows to group a set of observations into a given number $k$ of clusters. It relies on a vectorial representation of observations, in our case the derived vectors $V_s'$, for each session $s \in S$ and for a similarity measure among them, in our case computed based on the Euclidean distance, as described previously. Given a random initial selection of $k$ centroids for clusters, the algorithm operates by alternating between the two following steps, until no more changes are possible.

**Assignment step:** For each session, the algorithm computes the average distance between the session and all $k$ centroids. Then, the session is assigned to the cluster with the smallest distance.

**Update step:** Once all sessions are assigned to a cluster in the assignment step, the centroid for each cluster is updated. The new centroid is obtained by computing the mean for vectors that correspond to the sessions in the cluster.

#### 4- Selection of clusters

The $k$-means clustering algorithm allows to cluster video sessions into a given number $k$ of clusters. Multiple strategies exist for choosing the best $k$ value. In our work, we use clustering indices to compare multiple clustering solutions and choose the best one. For that, we run $k$-means algorithm for different values of $k$. A clustering index can then compare the different solutions and select the best value of $k$ for the final clusters. We combine the results of the following three indices.

**The Calinski-Harabasz (CH) index [17]** quantifies the dispersion level among clusters against within clusters dispersion. The best value of $k$ is considered as the one leading to the largest value of $CH$.

**The Silhouette (SI) index [18],** for a single data point, in our case a session, allows to measure how similar it is to the cluster where it belongs compared to other clusters. The average value of the Silhouette index over all the sessions indicates the consistency level in the clustering. The larger the value, the better the clustering.

**The Davies Bouldin (DB) index [19]** evaluates consistency in a clustering solution. It measures the within-cluster distances against between-cluster distances. A lower value of the DB index translates into a better clustering.

**Combining the indices:** To combine the outcome of the $CH$, $SI$ and $DB$ indices to find the proper number of clusters, we follow a ranking method. For each index, we assign a rank to each value of $k$. The value of $k$ that has the highest aggregated rank, according to the different indices is considered to represent the best clustering solution.

## V. Evaluation

We summarize here our assessment of our framework, starting with a presentation of the dataset we utilized. The clusters of sessions that the framework generated are described next.

#### 1- Dataset

The dataset we used to evaluate our framework was collected over the infrastructure of a real-world CDN provider. This CDN provider offers both VoD and live video content. The dataset encompasses sessions covering the transfer of these two types of content at a country-scale level. The initial dataset was collected for 14 days in 2016, with tens of thousands of content requests received on a daily basis. For each content request, the transfer of each of its chunks was tracked through data logs, with diverse information relating to the client, the content and the streaming node. For our evaluations, we operate over a subset of 6000 sessions occurring on a typical working day. For our evaluations, we considered two major KPIs, the Download Bit Rate (DBR) and the Quality Level (QL). The DBR is the rate at which bits are transferred from the surrogate server to the user. The QL represents instead the bit rate at which the video is encoded. Depending on the streaming technique used, the QL can change over time according to a user's requests (e.g. as in the case of adaptive bit rate streaming). Aside from being available in the dataset, these two KPIs have been selected for our evaluation because of their correlation with users' QoE, as shown in previous studies ([20] and [6]). However, our framework is generic enough to account for other KPIs that can be collected over the CDN system.

#### 2- Sessions clusters

We now discuss the outcome of our framework after its application to the obtained dataset. We start from the number of clusters selected. We recall that to obtain the best number of clusters, we rely on different clustering indices with multiple runs of the $k$-means clustering algorithm. We considered cases ranging from two to nine clusters. Among these options, the different clustering indices ranked the case of nine clusters as the best choice. Therefore, in the rest of the section, we analyze the corresponding nine clusters. We first examine the sessions that are grouped into the same clusters. We portray in Fig. 2 the evolution of the DBR and the QL over two sessions selected from cluster 4 and cluster 2, respectively. For cluster 4, we can see in Fig. 2(a) and Fig.2(b) that the two sessions present a similar evolution in terms of DBR and QL, with strong upwards variations in DBR and a constant QL.

In turn, the sessions selected from cluster 2, shown in Fig. 2(c) and Fig. 2(d), present similar DBR and QL variations; each is characterized by a single prominent peak for DBR and a constant QL with a few drops at the beginning. These observations illustrate our framework's ability to identify similarity in the KPI patterns between sessions.

Moreover, by comparing the sessions in Fig. 2, we can observe that the sessions in cluster 4 are significantly different from those in cluster 2, which also shows the capability of our framework to separate sessions with distinct KPIs patterns into different clusters. Similar observations also hold for the other clusters. We observed that sessions grouped inside the same cluster present very similar patterns in terms of DBR and QL evolution, while the sessions in different categories are clearly distinguishable. Furthermore, the identified patterns are informative for CDNs on users QoS and QoE, as follows.

In Fig. 3, we portray the evolution of the DBR and QL for a sample session from each cluster. The sample session is the closest session to the centroid of the cluster and is therefore representative of the patterns in the corresponding cluster. As can be seen, each cluster presents distinct patterns for KPIs evolution. Cluster 3 has strong variations in both DBR and QL. This reflects unstable network conditions throughout the video streaming, leading to frequent switching in QL of the viewed content. These frequent switches can significantly degrade the users QoE [21]. It is thus critical for the CDN provider to identify this pattern. A similar behavior is perceived in Cluster 5, with variations in both DBR and QL. However, in this case, variations are less prominent. Important variations in DBR and QL are also observed in Cluster 0. Nevertheless, a single high peak in DBR is enabling the distinction of corresponding sessions in a separate cluster.

Clusters 1, 2 and 8 present a few QL drops with distinct DBR patterns. Cluster 2 shows a constant QL with a few drops at the beginning and a single peak of DBR at the end of the session. The low QL values at the beginning of a streaming session are important to analyze as they have a significant influence on user's decision to carry on a streaming session [21]. Similarly, Cluster 1 is characterized by a few drops in QL together with variations and drops in DBR. There, the drops in QL and DBR at the end of the session are especially critical, as they could have resulted in the user abandoning the session. Similarly, cluster 8 presents a few drops in QL and some notable drops in DBR. Particularly important are the DBR drops that go below the stable QL and can reach zero. Such drops can lead to an undesirable video stalling behavior that affects users' QoE [21].

In contrast to the other patterns, Clusters 6, 4 and 7 present a constant QL, including sessions with constant bitrate streaming. Cluster 6 is characterized by variations in DBR with peaks at its session's starting and ending points. However, as the DBR values remain greater than the constant QL, based on these two KPIs, the video streaming would go smoothly. Both Cluster 4 and Cluster 7 exhibit a constant QL with upward variations in DBR that are more prominent in Cluster 4. However, both these clusters also present DBR drops below QL levels that can lead to interruptions in video streaming.

Overall, we notice the framework is capable of identifying meaningful clusters of sessions, with distinct KPIs patterns, that are informative for CDNs on users QoS and QoE.

## VI. Conclusion

In this work, we introduce a framework for the analysis of KPIs in large-scale CDN systems. The framework employs an unsupervised machine learning algorithm, to automatically form clusters of video sessions, presenting similar evolution of KPIs. We evaluate the framework over a real-world dataset. The results underline its ability to form meaningful clusters. In the future, we will expand the set of KPIs, features considered and clustering algorithm used. Moreover, we plan to extend the framework with QoS and QoE evaluators that determine the QoS and QoE in each cluster, with respect to the common KPIs patterns there. For clusters with low QoS and QoE, the evaluator will further conduct a root cause analysis by correlating observations inside the clusters.

## References


[1] Wang et al. "An overview of Cloud based Content Delivery Networks: Research Dimensions and state-of-the-art." *Transactions on Large-Scale Data-and Knowledge-Centered Systems XX*. Springer Berlin Heidelberg, 2015. 131-158.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2015–2020," White Paper, Jun. 2016.

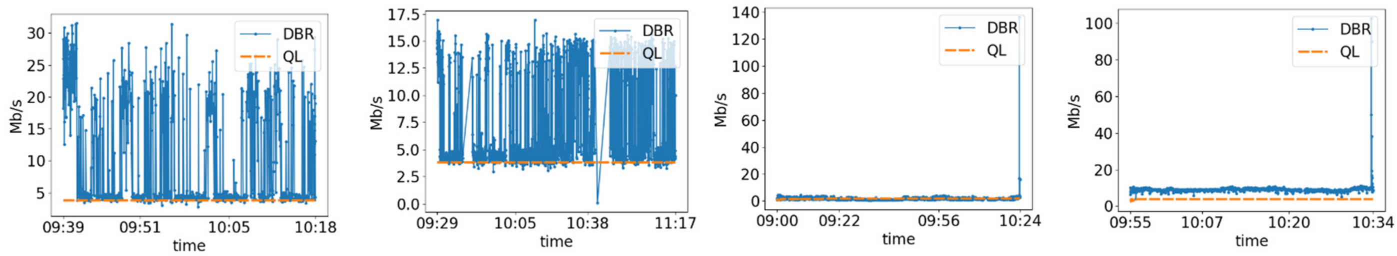
[3] Casas et al. "YouTube & facebook quality of experience in mobile broadband networks." IEEE Globecom 2012

[4] Fan et al., "Assessing Affinity Between Users and CDN Sites," *Traffic Monitoring and Analysis* 2015.

[5] Casas et al., "When YouTube Does not Work;Analysis of QoE-Relevant Degradation in Google CDN Traffic," *IEEE TNSM.*, vol. 11, no. 4, Dec. 2014.

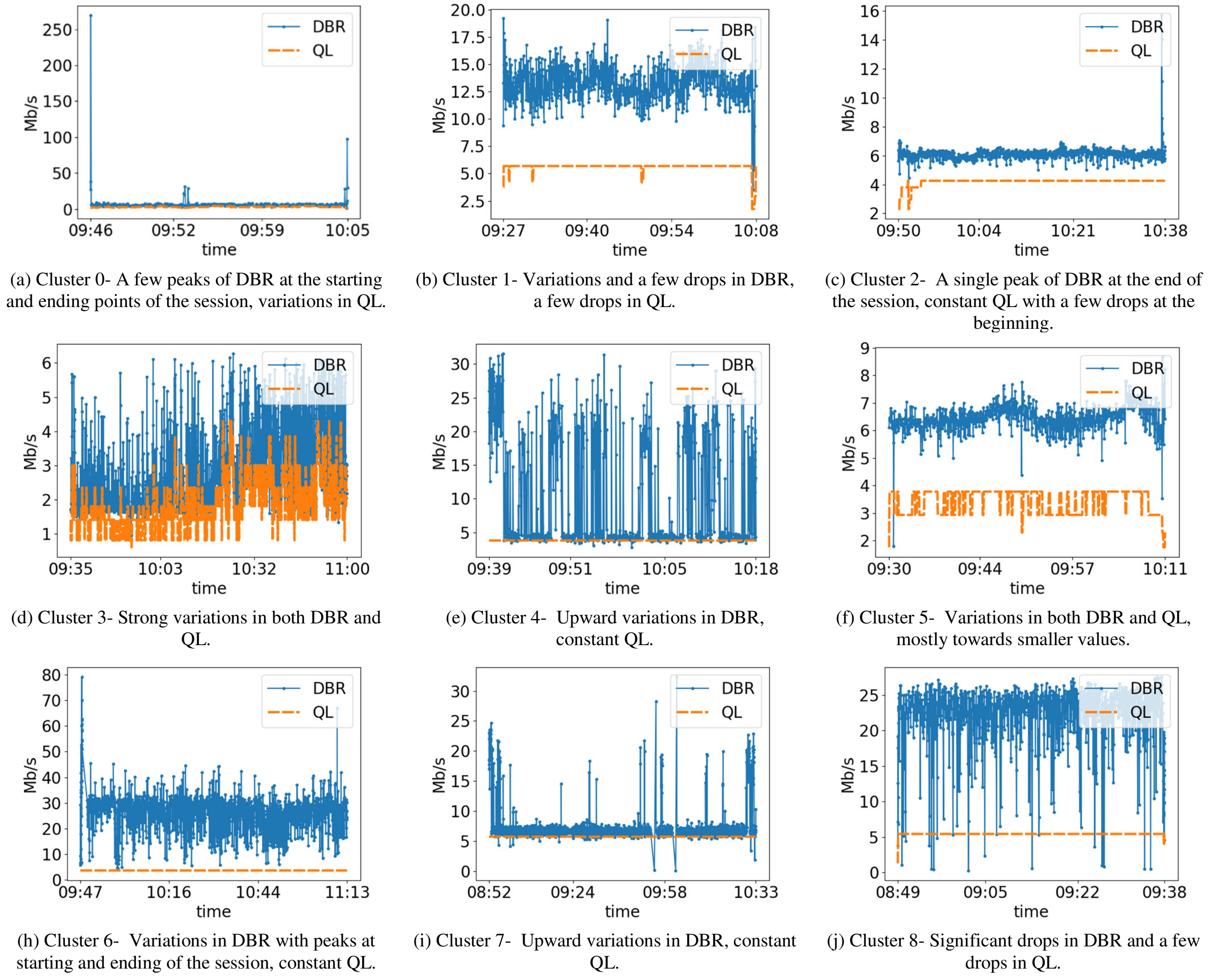
[6] Shafiq et al., "Understanding the Impact of Network Dynamics on Mobile Video User Engagement," *ACM SIGMETRICS*, New York, NY, USA, 2014, pp. 367–379.

[7] Li et al., "Video Delivery Performance of a Large-Scale VoD System and the Implications on Content Delivery," *IEEE Trans. Multimed.*, vol. 17, no. 6, Jun. 2015.


(a) (b) (c) (d)

Fig. 2. Sessions (a,b) in cluster 4; Sessions (c,d) in cluster 2.




(a) Cluster 0- A few peaks of DBR at the starting and ending points of the session, variations in QL.

(b) Cluster 1- Variations and a few drops in DBR, a few drops in QL.

(c) Cluster 2- A single peak of DBR at the end of the session, constant QL with a few drops at the beginning.

(d) Cluster 3- Strong variations in both DBR and QL.

(e) Cluster 4- Upward variations in DBR, constant QL.

(f) Cluster 5- Variations in both DBR and QL, mostly towards smaller values.

(h) Cluster 6- Variations in DBR with peaks at starting and ending of the session, constant QL.

(i) Cluster 7- Upward variations in DBR, constant QL.

(j) Cluster 8- Significant drops in DBR and a few drops in QL.

Fig. 3. Summary of obtained clusters.

[8] Li et al., "Understanding the relationships between performance metrics and QoE for Over-The-Top video," in *2016 IEEE ICC*, 2016, pp. 1–6.

[9] Orsolic et al., "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Multimed. Tools Appl.*, pp. 1–35, May 2017.

[10] Giordano *et al.*, "YouLighter: A Cognitive Approach to Unveil YouTube CDN and Changes," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 2, pp. 161–174, Jun. 2015.

[11] Wu et al., "Scalable network-based video-freeze detection for HTTP adaptive streaming," *2015 IEEE IWQoS*, 2015, pp. 95–104.

[12] Dimopoulos et al., "Identifying the Root Cause of Video Streaming Issues on Mobile Devices," in *Proceedings of the 11th ACM CoNEXT*, New York, NY, USA, 2015.

[13] Zhu et al., "LatLong: Diagnosing Wide-Area Latency Changes for CDNs," *IEEE TNSM*, 9(3), Sep. 2012.

[14] Liao et al., "Clustering of time series data—a survey," *Pattern Recognit.*, vol. 38, no. 11, Nov. 2005.

[15] Teemu et al. "Feature-based clustering for electricity use time series data." Adaptive and Natural Computing Algorithms, 401-412, 2009.

[16] Xu et al., "Survey of clustering algorithms", IEEE Transactions on Neural Networks, 16(3):645–678, 2005..

[17] Caliński et al., "A dendrite method for cluster analysis," *Commun. Stat.*, vol. 3, no. 1, Jan. 1974.

[18] Wang et al., "CVAP: validation for cluster analyses." Data Science Journal, 88-93, 2009.

[19] Davies et al., "A cluster separation measure." IEEE transactions on pattern analysis and machine intelligence 2, 1979

[20] Liotta et al., "Quality of experience models for multimedia streaming," Int. J. Mobile Comput. Multimedia Commun., vol. 2, no. 4, pp. 1-20, 2010.

[21] Juluri et al., "Measurement of quality of experience of video-on-demand services: A survey." IEEE Communications Surveys & Tutorials 18.1 (2016)