*Chapter 1*

# Big-Data helps SDN to improve application specific quality of service

*Susanna Schwarzmann, Andreas Blenk, Ognjen Dobrijevic, Michael Jarschel, Andreas Hotho, Thomas Zinner, Florian Wamser*

## 1.1 Introduction

Managing the quality of real-time multimedia services, such as video streaming and networked virtual reality, still poses many technological challenges. For instance, data rate demand of video streaming services is dramatically increasing. At the same time virtual reality applications call for low user-to-server latency. These opposing demands are dictated by the evolution of the quality concept, which has been transformed over the past decade from more technical, network-level Quality of Service (QoS) into user-centric Quality of Experience (QoE) [1, 2]. Going beyond QoS, which commonly conveys network performance in terms of measurable parameters like throughput and delay, QoE identifies additional factors that influence service quality as perceived by end-users. The latter QoE influence factors (QoE-IF) may, for example, include user-device screen resolution and previous service usage experience.

Recently, many research results have exploited the paradigm of software-defined networking (SDN) [3] as a means to implement QoS-/QoE-oriented network control and management (CaM). The respective "CaM loop" aims at customizing the network configuration to reflect the specified quality improvement target, e.g. reducing the number of video stream stalling events. SDN, with its separation of network control logic from data plane devices into distinct controller entities, provides architectural blocks to realize QoS-/QoE-centric CaM [4]. Well-defined communication interfaces, as advocated by SDN, enable network applications that work with SDN to communicate information about multimedia application states to the network controller, on the one hand. On the other hand, SDN facilitates the acquisition of network-wide performance statistics on the controller entities by means of, for example, OpenFlow [5]. As a result, SDN is able to maintain a global application state across the network, supported by the fact that it has access to both stakeholders.

Furthermore, with the introduction of QoE as a CaM objective, which calls for (a) measuring and collecting QoE-IF data, (b) processing and analyzing this data, and (c) producing and enforcing action decisions, network CaM faces the challenges of dealing with large data sets, i.e. "Big Data" (BD) [6]. Recent network-level solutions were only provided for small data-scale scenarios, since they lack BD-related technologies that are able to handle huge data sets. In the QoE CaM context, a respective SDN system might consider a vast number of QoE -IF data sources (i.e., *data variety*) that produce large data quantities (or, *data volume*) on different time scales (referred to as *data velocity*). New developments of BD technologies, e.g. Deep Learning or MapReduce, allow for an efficient processing of such large data. Moreover, BD techniques facilitate efficient execution for most of the state-of-the-art machine learning and data mining algorithms. Combining SDN control logic with methods of BD analytics, e.g. by integrating them into an SDN controller, would enable taking into account a wide range of QoE-IFs and, thus, "more precise" decision-making that conforms to the specified CaM goal. Moreover, BD techniques could be used, e.g., for customizing QoE estimation models during service run-time to consider categories of end-users with different demographics.

This chapter first provides an outline of the current results in the domains of: (a) QoS/QoE CaM for real-time multimedia services that is supported by SDN, and (b) BD analytics and methods that are used for QoS/QoE CaM. Then, three specific use case scenarios with respect to video streaming services are presented so as to illustrate the expected benefits of incorporating BD analytics into SDN-based CaM for the purposes of improving or optimizing QoS/QoE. In the end, we describe our vision and a high-level view of an SDN-based architecture for QoS/QoE CaM that is enriched with BD analytics' functional blocks and summarize corresponding challenges.

## 1.2 Classification of SDN-based Context-aware Networking Approaches

In the following, we discuss various approaches that use SDN for QoS/QoE-oriented network control and management of multimedia services (QoS/QoE CaM). Since, in particular, *data variety* and information gained by monitoring, as well as *data analytics* and related control actions play an important role in the Big Data context, we consequently classify the presented approaches based on QoE influence factors (QoE-IFs), the control actions triggered from these information, and the resulting implications. An overview of the investigated approaches with their classification is shown in Table 1.1.

In the area of SDN and QoS/QoE management, video streaming is currently one of the main drivers, as it is responsible for most of the Internet traffic [7] and plays a strong role with representatives such as MPEG DASH and HTTP adaptive streaming solutions. For that reason, the solutions discussed in the following all focus on video streaming or video conferencing.

| Source | Monitored QoE-IFs | Control Action | Implication |
|---|---|---|---|
| [8] | Packets in the network (DPI) | App-aware path selection | Prevention of video stallings |
| [9] | Packet loss, delay | App-aware path selection | QoE-enhancement of a multitude of applications |
| [10] | Network congestion indication | Flow prioritization | Quality enhancement of live video transmission |
| [11] | YouTube Video Buffer | App-aware path selection | Prevention of video stallings |
| [12] | Network throughput, video buffer | Flow prioritization and quality adaptation | Prevention of video stallings |
| [13] | Active DASH streams, available bandwidth, client properties | Bandwidth reservation | Fairness w.r.t. video quality |
| [14] | Available bandwidth, network latency, client properties | Dynamic resource allocation | Fair QoE maximization |
| [15] | Available bandwidth, packet loss rate, jitter, initial delay, buffer | Change of routing paths and transport nodes | QoE enhancement for video streaming |
| [16] | Active DASH streams, network resources, client properties | Bitrate guidance, bandwidth reservation | Fairness w.r.t. video quality |
| [17] | Available bandwidth, video buffer | Network resource allocation | Fair QoE maximization |

Table 1.1: Classification of SDN-based context-aware networking approaches

### *1.2.1 Monitoring of QoE Influence Factors (QoE-IF)*

We begin with a classification of selected management approaches according to the monitored parameters. Each SDN-based CaM approach monitors at least one QoE-IF. These factors can be classified in terms quantity and location of monitoring. We consider in the following four dimensions of monitoring. There are commonly mechanisms that perform (1) monitoring in the network at packet level, (2) monitoring at flow level, (3) monitoring of application information that are available within the client software, and (4) mechanisms that perform monitoring at network and application-side, i.e. monitoring of network parameters as well as application-side QoE-IFs updates.

**Monitoring at packet level in the network.** An approach that exploits network information on packet level is [8] by Jarschel *et al.* for QoE management of web and video traffic. By means of Deep Packet Inspection (DPI), packets are inspected on their way from source to destination. Based on significant fields in the packets the application can be identified. The challenge is that a wide range of information has to be collected at different points in the network in order to get a holistic picture of the application in the network. It also discusses whether a northbound interaction between application and SDN controller is more beneficial than packet DPI, since encrypted traffic poses another major challenge through end-to-end encryption. The approach from [9] collects information about packet loss and packet delay and refrains from the reading of application information from packet payload. A major challenge is here the collection of packet-level statistics at different points in the network. The efficiency of such approaches and the associated detection of the bottlenecks in the network strongly depends on the possibility of comprehensively monitoring the network.

**Monitoring at flow level in the network.** Following the approaches that monitor the network on packet level, there are also approaches that examine the network at the flow level. For this purpose, no individual packets are analyzed, but the flow through the network is considered. In [10], for example, monitoring is performed to quickly detect network congestion based on network flow statistics.

**Monitoring of application information.** An example where application information is used as basis for control actions is [11]. This proposal relies on the client's buffer state as QoE-IF. In this case, the client buffer state is only one example, which is investigated for the considered use case video streaming. A further development of collecting information on application level might reveal that a targeted monitoring of specific application parameters is desirable for each active application and for each client to optimize QoS/QoE management in the network.

**Monitoring at network- and application-side.** Unlike the previously mentioned mechanisms, which either rely on network monitoring or application monitoring only, the mechanisms presented in [12–17] consider both, application and network information, to decide about control actions. This may include upon others the number of active video streams, the capabilities of the used devices in terms of screen resolution, and QoE-IFs like current buffer or the number of quality switches. Thereby, a new challenge arises in multi-application scenarios, where information on other applications' traffic flows must be gathered as well at a large scale. This requires monitoring variety of data, but it also offers a value-added, as it is possible to make decisions for the benefit of all applications. All above-mentioned challenges lie in the direction of Big Data and machine-based data analysis, because data variety and volume are crucial for the success of these approaches.

### *1.2.2 Control Actions of Management Approaches*

The presented mechanisms are based on various adaptations in the network in order to meet the requirements of an application. In addition to the control actions implemented in the network, some of the approaches also take into account additional control actions within the application. However, in the following, we are only going to detail the network-side adaptations, since the focus in this work is on SDN, which naturally performs the adaptations on the network side using the OpenFlow Southbound API. As the presented classification table shows, several proposals employ the same or similar control actions. Dynamic application-aware path selection is performed in [8, 9, 11], and [15], whereby the latter one additionally performs a dynamic selection of the transport node. Having the knowledge from the monitoring entities and the QoE models, an algorithm decides about the network path for the specific flows in order to meet the application requirements. The challenge in this context is to make coordinated, fine-grained decisions. The granularity of the information hereby improves the decision-making process. Fine-grained information can be used to carry out more targeted actions on the network. If also machine learning is used on the mass of data, it is possible to better estimate the required application parameters influencing the control actions, such as clients' video buffers. For all approaches, the amount of data is essential to make efficient decisions and not to discriminate against other applications on the network.

The mechanisms of [10, 12] temporarily prioritize specific flows in the network in order to prevent QoE degradation. This is realized by implementing at least two queues, whereby one is set up as best-effort queue, while the other one processes the packets of the prioritized flows. The packets in the high-priority queue are preferably scheduled in contrast to the packets in the best-effort queue.

Dynamic allocation of resources, e.g. bandwidth reservation, is considered in [13, 14, 16, 17]. These mechanisms have in common that they take into account fairness aspects. This can either mean that all video clients - which possibly have different device capabilities (e.g. screen resolution) and hence different demands on the network - have a fair video quality, or that the QoE is maximized whilst fairness constraints are considered.

Big Data and machine learning approaches have potential to support those mechanisms in the decision process. For example, it is conceivable to use reinforced learning. An algorithm learns from the impacts triggered by specific control actions. Hence, it continuously optimizes its decisions and is aware of the currently best-fitting control action. The basis on which the algorithm decides on control actions, i.e., the feature set considered for learning, is extensive and includes, among other things, commonly network and application behavior. By using user-defined data obtained through monitoring the user behavior (e.g. video-stalling duration or initial delay thresholds that provoke the user to abort), it is even possible to react in a user-centric manner.

### *1.2.3 Potential of Big Data for SDN QoE Management*

The applicability of Big Data for SDN-based QoE management approaches is indisputable as discussed in the previous sections. The trend for more data and more monitored QoE-IFs dictates the use of Big Data in this area. Nevertheless, at the moment approaches do not exploit this potential and avoid the use of Big Data, since the approach and the way of thinking are different with this mass of data.

In contrast to the traditional approaches, Big Data helps in evaluating information in three different directions. Firstly, Big Data supports the statistical analysis of encrypted traffic, which is important in today's networks. On the basis of privacy issues traditional approaches refrain on packet analysis and rather collect statistics on network and packet throughput to get information about the applications. Secondly, Big Data can help to analyze data within the whole network at different points of presence. Many CaM approaches can improve their optimizations by taking into account information about the entire network. Thirdly, Big Data also helps at application level where all applications need to be considered and, consequently, a lot of information needs to be gathered.

Besides the collection and analysis of the huge amount of data, BD can help with its analytics methods. Feasible are, for example, algorithms that facilitate to learn and predict appropriate control actions based on the given data. Some ongoing work in the context of Big Data and QoS/QoE management is presented in Section 1.3.2, including also examples where network control decisions rely on the outcomes of Big Data analytics mechanisms.

The challenge in the context of control actions is to make coordinated, fine-grained decisions. For all approaches, the amount of data is essential to make efficient decisions and not to discriminate against other applications on the network with respect to fairness in the network.

### *1.2.4 Conclusions*

We conclude this discussion about SDN-based QoE management approaches and the related assessment for the approaches with respect to Big Data with a list of challenges:

1. **Encrypted traffic represents a challenge for traditional SDN QoE management approaches.** Encrypted traffic makes it difficult to use DPI procedures. Instead, more statistical methods based on a lot of data need to be used. Patterns in traces can help to train models to specify application classes although traffic is encrypted.

2. **Network-wide monitoring is a challenge for QoE management approaches.** A network-wide overview of key QoE-IFs is necessary for efficient and fair control decisions. To control the network (e.g. path selection), it is necessary to know the complete network. This requires lots of information from lots of devices throughout the whole network.

3. **At the application level, the challenge is to monitor all applications with appropriate granularity.** When monitoring is performed on application layer (e.g. to support QoE fairness), it is not sufficient to monitor one client or application instance. You must be aware of all relevant applications running and their requirements.

4. **Dedicated network control is a challenge.** For fine-grained and targeted control actions in the network, information from all areas must be known. For

all approaches, the amount of data is essential to make efficient decisions and not to discriminate against other applications on the network with respect to fairness in the network.

5. **Other challenges.** Another further challenge for management approaches is the processing of the large amount of data. The questions arises how to store, handle and structure the different information with respect to the desired outcome. Additionally, subjective studies on QoS/QoE mapping, which are needed to train models to automatically identify the resulting QoE, are very costly.

The key derivation must therefore be, that in order to counter the challenges of QoE management in the present time, the network and application status must be learned and the effects of the actions on the network must be examined. Next, the resulting model must be set up with the help of unsupervised or supervised learning methods to automate network/application control actions in an efficient way.

## 1.3 Big Data Analytics to Support QoS/QoE Management

In this section we focus on the potential of Big Data analytics to support QoS/QoE management. We first give a short overview on BD analytics approaches and afterwards present current work that apply those techniques in the context of QoS/QoE management.

### *1.3.1 Big Data Analytics*

This subsections provides a short overview on typical Big Data analytics techniques. We will not discuss BD in general as it focuses not only on machine learning and data mining approaches but also addresses a broad range of data handling aspects [18]. Data handling aspects are only of limited importance here since we have to deal with the "3 Vs": *Volume*, we need to handle a huge quantity of data, *Velocity*, as we need to deal with the incoming data just in time and partially with *Variety*, when we bridge the gap between the network flow data and the application level. We ignore the other "2 Vs" of Big Data. Therefore, we will focus on typical machine learning and data mining approaches which form the basis for an analysis of the collected data with special emphasis on BD aspects. An in depth discussion of the combination Big Data and SDN can be found in [19].

The goal of machine learning (ML) is to learn from a given set of examples and to build models from it. This model can be later applied on newly and unseen data. A second goal is to gain new insights about present data by means of those models. Data Mining includes this model learning step in a bigger process, which includes also data handling and application of learned models as other important steps. Due to this data centric view, a bunch of new "data mining" techniques have been developed in the past. The most prominent example is the association rule mining approach which is part of a more general class of methods known under the term *pattern* *mining*. In general, ML and DM techniques are broadly classified into *Supervised-*, *Semi-supervised-*, or *Unsupervised Learning*. An introduction to Machine Learning can be found in [20–22] and to Data Mining in [23–25].

When supervised learning is applied, the classification rules (model) are learned based on labeled data. Labeled data, or training data, indicates the desired output, the correct feature value, depending on the given input. Hence, the model builds a relationship/function that relates input parameters to the output feature. This model is then applied to unlabeled data and the output is predicted based on that. Typical supervised learning algorithms are Support Vector Machine (SVM), Decision Tree, Naive Bayes, k-Nearest Neighbor (k-NN), and Random Forest. These techniques are often super-ordinated as *Classification*. In contrast to classification, where the output variable is a pre-defined class, *Regression* predicts continuous values. Other learning approaches like bagging, boosting or ensemble learning combine either weak or strong learners to a new model.

The term *Clustering* denotes the unsupervised learning methods, where no labeled data is given in advance. Typical clustering techniques include density based methods like DB-Scan, standard statistical approaches like k-Means, k-Medoids, and Expectation-Maximization (EM) (for a survey [26]). More recently, methods like LDA become popular in many areas. The basics are already addressed in text books like [22].

Semi-supervised learning is part of supervised learning, with the difference that it makes use of both, labeled and unlabeled data. In this way, fewer labeled data is needed, but as larger quantity of the available data is used, a more general model can be learned.

Machine Learning and Data Mining approaches discussed so far typically need to be able to access all data during the model learning or pattern detection phase. Storing all the “Big” data is sometime impossible and therefore, classical ML and DM methods can’t be applied. Stream data mining refers a set of methods adopted in such a way that models can be learned or patterns can be detected directly from a stream of data. Besides adopted standard methods like tree learner, one can find special methods like time series analysis which inherently rely on data streams in this area. An overview of stream mining algorithm is given in [27]. As there is no longer a need to store the data, we can address the Big Data issue of Volume and Velocity within our network analysis setting. We can directly stream the data to a learning machine which computes new models on the fly. These models can be deployed on the network devices or controllers and take care of the network flow. The Apache Storm framework[1] provides a distributed stream processing framework which can be adopted to efficiently learn from a data stream.

Reinforcement learning is another machine learning paradigm (cf. [28]) where an agent autonomously learns a strategy. It can be seen as a kind of weak supervision, as minimal feedback is provided which is used to learn the strategy. The agent is not trained in terms of actions to take, instead, it is rewarded (positively as well as negatively) for its decisions. Typical examples are game playing (the feedback is winning the game or getting higher score) or controlling machines like a robot mov-

[1] http://storm.apache.org

ing trough a labyrinth. The benefit of such a system is, that it learns continuously, even if it is in practical application.

With the advances of deep learning methods, reinforcement is becoming more and more popular and successful as shown by examples like game learning for Atari computer games [29]. The main idea is the use of a deep neural network to do the different processing steps on the corresponding layer. This includes the image processing which in a classical learning step would be a kind of feature engineering, but also the judgment of the reward over the long time. Without going into details of how deep learning methods work, one could apply similar deep learning approaches on network traffic with the goal of controlling the flow trough the network. The neural network could learn the reconfiguration of the network by directly analyzing the network traffic. This could also be done in a stream stetting, by utilizing one of the stream frameworks.

If we are not able to do stream mining we need to store and process BD for learning. In the past years a bunch of typical paradigms were developed. Among them are the MapReduce approach implemented in Hadoop (disk focused) or Spark (memory focused) mainly developed by the search engine vendors to process web scale data. MapReduce is a method to efficiently process large datasets [30]. The two functions *map* and *reduce* form the key of the approach and call each other iteratively. During the map phase, the input data is filtered or sorted with respect to some criteria implemented as a user-defined function in parallel. The results are distributed and send to the reducers. The reduces summarize the values in order to obtain a smaller set or even the final result set and return it. If the data is stored in a distributed fashion the first map job will directly access these distributed data which allows us to easily work on a big network dataset in parallel. Another paradigm developed in the past years to store huge amount of data are NoSQL databases. In contrast to classical relational databases, these databases follow a different main principles when storing the data, like columns, documents, key-values, graphs, and multi models. Beside the change of the storage model, such databases favor speed over traditional properties like consistency. An introduction into the new often distributed storage models can be found in [18].

### *1.3.2 Current and Ongoing Work*

Mestres et al. [31] present a new paradigm called Knowledge Defined Networking (KDN) based on the idea of a Knowledge Plane for the Internet [32]. Their idea is to learn from network behavior and automatically operate the network accordingly. They present a loop of constant learning. An SDN controller analyzes the network and provides the information to an analytics platform that transforms this information into knowledge. To do so, several ML techniques are applied: Supervised learning, unsupervised learning, and reinforced learning. The knowledge is provided to the controller, which can find appropriate control instructions based on this knowledge and its global network view. Information about performed control actions and impact on network behavior are again provided to the analytics platform. The authors present two use-cases for the proposed Knowledge Plane. The first one focuses routing in an overlay network, the second one targets on resource management in an NFV scenario.

In [33], six classifiers (Naive Bayes, SVM, k-NN, Decision Tree, Random Forest, and Neural Networks) are compared with respect to their applicability to estimate the QoE from QoS parameters. The authors present a framework in which users can rate their satisfaction with the quality of a YouTube video during playback and after the playback is completed. Simultaneously, the framework monitors the video characteristics (QoS parameters). The framework is used within a large-scale crowd-sourcing study in order to obtain training data which map video QoS to QoE values. Besides the crowd-sourced approach, the authors conduct experiments in a controlled environment. Hence, the objective MOS can also be matched with QoS parameters like packet loss, jitter, and delay. Based on this data, models have been trained for the six different classifiers. With regard to the mean absolute error, Decision Tree (DT), yields the best classification result in a 4-fold cross-validation. In terms of the correctly classified share from the test set, Random Forrest and Decision Tree outperform the other techniques.

A methodology for estimating YouTube QoE based on statistical properties of encrypted network traffic is presented in [34]. The authors set up a testbed where several YouTube videos are played back. During playback, the network traces are stored. These traces provide information like packet length, size of transferred data within a fixed interval, packet count statistics, and TCP flag count. Further, application-level data is captured during video playback. This includes the number of stallings, stalling duration, and playback time on a certain quality level. Based on these QoE-related parameters, each video instance is classified into one of three QoE-classes: low, medium, and high. Several experiments with varying video durations and bandwidth configurations resulted in 1060 videos in total, including the associated network traces. Using WEKA, this data is used for feature selection and model building with several classifiers (OneR, Naive Bayes, SMO, J48, Random Forest). Again, Random Forest outperforms the other methods w.r.t. accuracy when the model is trained and tested using 10-fold cross-validation of the whole dataset.

Traffic classification is also targeted in [35]. However, unlike the previous approach, the authors do not aim at predicting a QoE value, but at classifying network traffic into one of several QoS-classes. A QoS-class summarizes applications that have similar QoS requirements, e.g. voice, video conference, streaming, bulk data transfer and interactive data are considered as QoS-classes. To learn the classifier, network traces are stored and labeled as one of these classes. The knowledge about an application's QoS class can then be used to perform a QoS-aware traffic engineering in order to satisfy the application's needs.

The authors propose to apply the classifier within a framework that is located in an SDN controller to take advantage of its global network view, programmability, and computation capacity.

The feasibility of different ML algorithms for traffic classification is investigated in [36]. The authors use the OpenFlow protocol to gather information about the traffic in an enterprise network. They store several features of the TCP flows and the corresponding packets. These features include flow duration, packet time stamp, inter-arrival time and packet count. To obtain labeled data, they run applications in a controlled experiment and store the traffic traces caused by the different applications. This data set is used to train models for predicting applications based on network data with three different classifiers: Random Forest, Stochastic Gradient Boosting, and Extreme Gradient Boosting. Their results indicate that each of this supervised learning techniques can obtain a high traffic classification accuracy.

Statistical regression analysis is used in [37] to determine the relationship between several QoS parameters and the resulting QoE for video conferencing on MOS scale. The authors consider upon others packet loss rate, round trip time, bandwidth, and jitter to produce the regression coefficients. These coefficients are analyzed for several access technologies (e.g. Wi-Fi and 3G) in order to predict the QoE for several access technologies with the goal to dynamically select the technology providing the best QoE.

Two more approaches for estimating Qoe from QoS parameters are presented in [38, 39]. The authors propose to use the predictions to find the input network parameters to obtain a QoE that satisfies a user's needs and to decide about appropriate network management.

The user QoE in an enterprise working environment is focus in [40]. The authors evaluate the potential of several machine learning algorithms to predict the worker satisfaction based on objective measurements (waiting times). They use results from a subjective user study and technical data from the system monitoring to learn three models, namely SVM, Gradient Boosting, and Deep Neural Networks. The resulting classification accuracies reveal that none of the examined algorithms is reliably applicable for QoE prediction based on non-intrusive application monitoring data. However, when modeling on a per-user scale, there is a share of about 5-10% of all users, whose models can classify with over 80% accuracy. Hence, the QoE may be predicted with good performance for specific users if personalized prediction models are applied.

## 1.4 Combining Big Data Analytics and SDN: Three Use-cases to Improve QoS/QoE

In this section, we present three use-cases which illustrate the envisaged benefits of combining BD and SDN. The first use-case is an extension of classical network QoS-monitoring to achieve improvements and adjustments in the network due to certain network settings. The second use-case assumes a business agreement between a video on-demand provider and an SDN-based network operator to exchange values of QoE-IFs, which are then processed by BD-applications. The final use-case, as opposed to the second one, assumes no direct communication between the video service provider and the network operator, while Big Data applications are utilized in order to infer the service-level QoS/QoE.

### *1.4.1 Improving the Operation of Networks*

This subsection deals with the use-case of improving network operation by combining Big Data and SDN. In particular, we discuss compliance of network performance with the QoS-requirements for Voice-over-IP (VoIP) traffic.

Traffic flows and their mutual influence within networks are highly complex and unpredictable in today's networks. Network-level actions, like queuing, traffic shaping, selective dropping and link-efficiency policies, provide a network operator with control over how these flows transition the network. This is especially critical for VoIP and video streaming traffic, since the operator needs to improve network operation and maintain the specified QoS requirements, such as maximum allowed latency and minimally required throughput. From a technical perspective, this means that in cases where network virtualization is not possible, or the use of technologies such as virtual local area networks (VLAN) is not adequate, network settings and QoS optimizations can be used in the network to enable a robust traffic flow.

A good example are VoIP networks. There, a telephony application typically requires the one-way latency not to exceed 400 ms [41]. This must apply to the entire network, if VoIP traffic is being transported. In this case, layer 3 markings (preferably Differentiated Services Code Point, or DSCP) or layer 2 prioritization with the Class of Service (CoS) markings are commonly used for this purpose, in the outbound direction of each network link.

A continuous measurement and monitoring of the important quality features in the network forms the basis for the VoIP QoS-management. In the network, switches and routers are currently being used to generate NetFlow statistics on packet latency and to perform active tests on how to meet the current QoS requirements for VoIP. In terms of Big Data and Big Data Analytics, two general paradigms can be applied in addition to the traditional monitoring and testing: (1) the collection, storage and processing of the data on a high detail-level using BD mechanisms, and (2) the analysis and evaluation with BD learning methods to provide better insights, detect failures, predict future critical situations and usuage trends without direct operator interaction.

1. **Collection, storage and processing of the data according to the Big Data principles.** Through large-scale collection and storage of data, QoS-statistics can be collected across the entire network. It is even possible to add application information (*Variety*) as additional source to do a better network control. Big Data provides means for efficient data storage, e.g., NoSQL databases, how the storage cluster needs to be scaled based on the data volume, and how the data needs to be processed to meet analytical engines such as Hadoop. The new data allows not only for more detailed statistics due to the higher *Volume* of the data. Even more, these data are the basis for learning new models and extract the hidden knowledge about the usage patterns of the network. Due to the new size, the insights are more fine grained and the action will allow for more specific and timely (*Velocity*) reaction with respect to users' need. This could even reach a level where personalized traffic requests can automatically be met by the network when learning is used.

2. **Analysis and evaluation with Big Data learning methods.** By analyzing the collected data and learning QoS models from it, conclusions can be drawn about the QoS-compliance. It can be checked whether the QoS-requirements are enforceable or not, and whether the QoS should be adapted based on the models learned from historical data. Daily patterns and traffic situations can be estimated, appropriately handled, and evaluated for the network control purposes. It is even possible to predict future traffic situations and to take long term actions based on the collected data. Examples of successful machine learning applications on network traffic are described in Section 1.3.2 which show what is currently possible with state of the art models. With the adoption of stream data mining and deep learning models, we expect self-adaptable SDN controllers given some high level strategy of the network provider which show the full potential of Big data analytics in this area.

The implementation and configuration for QoS management takes place in the entire network with the help of SDN. With SDN, for instance, the control actions are passed to the devices and dynamic adjustments can be made based on the output of the BD-analytics engine.

In the end, the use of BD in QoS-management means the logical continuation of the idea, in which data is evaluated to enforce QoS-requirements for special types of multimedia services.

### *1.4.2 Improving the Quality of Video-on-Demand Streaming based on Business Agreements*

This subsection gives another example of how the integration of BD applications into an SDN-based network environment can enhance QoS/QoE. The example assumes that a video-on-demand (VoD) streaming service provider (SP), e.g. Netflix or Amazon Prime, has negotiated with a future SDN-based network operator (SNO) to exchange service-level and network-level information relevant to QoS/QoE control. Such a business agreement between SPs and SNOs may provide mutual benefits: SPs offer improved QoS/QoE to their end-users, while NPs can utilize their network resources more efficiently. The business agreement encompasses varying points. The SNOs agree to provide ”prioritized” traffic treatment for the SPs’ customers.

Further, SPs and SNOs agree on the exchange of values for the relevant QoS/QoE-IFs. In case of VoD streaming, we identify the following parameters to be reported by the SPs:

- (anonymised) user demographics data (e.g. user age range), which is reported during the video session establishment phase;
- previous service usage experience (beginner/advanced user), which is reported during the video session establishment phase;
- service cost (flat rate, cost per video, etc.), which is reported during the video session establishment phase;

- user device type (e.g. smartphone, tablet, and laptop), which is reported during the video session establishment phase;
- user device characteristics (screen size, OS, CPU and RAM features, etc.), which are reported during the video session establishment phase;
- video client statistics (e.g. buffer status, number of video freezes), which are reported periodically for the session duration;
- service features (MPD information), which are reported during the video session establishment phase; and
- server statistics, which are reported periodically for the session duration

However, this constitutes a large number of QoE-IFs, which need to be efficiently monitored and provided by SPs. Here, BD-applications might be utilized by the SPs in order to efficiently process and compress the monitored data on end-users and VoD service.

In order to put such an architecture into effect, further implementation steps are necessary. As a first adjustment, the VoD clients and servers would be extended so as to report QoS/QoE-IFs, e.g., by piggy backing HTTP traffic. For the information exchange between SPs and SNOs, an orchestrator may be used that serves as the collection and extraction point to the data on relevant influence factors. To interact with the SDN control plane, the orchestrator can use a Northbound interface provided by one of the open-source SDN controllers.

To make use of Big Data services, two ways are possible. Either the orchestrator interacts with a Big Data infrastructure via another interface, or it integrates Big Data applications directly. Furthermore, the operation of the orchestrator might be optimized on run-time with the help of Big Data applications. For instance, delay information might be extracted, which helps to improve the network optimization. In addition, BD-applications lower the burden of extensive data processing on SDN controllers, or even relieve them of the processing raw information completely.

On the SNO side, SDN controllers might periodically collect network-wide statistics on the performance of the data plane elements. Based on the received information from the orchestrator and the monitoring data on the network, the control plane can make the best possible decision according to the business agreement and the overall network optimization goal. The SDN controller could use VoD service and end-user information to make decisions. This information could allow the SDN controller to make distinctions, e.g., between advanced users and beginners ("a beginner is less likely to be annoyed with video flickers than an advanced user"). Other end-user information could be the service cost. Here, an end-user paying for each video expects more value-for-money than a flat-rate end-user. Such insights and metrics can be delivered via Big Data applications running on the SP side. Furthermore, other BD-received information could provide insights into the reasons of video freezes. Here, end-user information allows to differentiate between video freezes due to poor client performance, e.g. a stressed end-device running too many applications, and video freezes as a result of misconfigured network operations. Other service information can be frequent changes in video quality. Thus, the SDN controller can support video traffic to provide a more stable delivery and thus reduce quality changes. In case of general over-utilization, information retrieved from BD applications allow to distinguish end-users based on their previous usage experience, i.e., history of application use.

### *1.4.3 Improving the Quality of Applications without Business Agreements*

In a setup where no direct negotiation and information exchange between SNOs and SPs exist, Big Data applications can still help to improve the overall service and network performance. In this case, the incentive for an SNO is to serve end-users with the best possible network performance, as they would most commonly blame the SNO for poor service quality.

The challenge is to identify reasons for service performance degradations, in particular for encrypted network traffic. Thus, the goal of an SNO would be to establish a network monitoring infrastructure with an SDN controller that is making decisions based on the efficiently monitored data. In order not to burden the controller with intensive data processing, the monitoring infrastructure might incorporate BD techniques. BD applications would then provide statistics of video streaming traffic based on, e.g., average packet size, inter-arrival packet time and average throughput. Since this again might be a huge amount of data, it would be of immense importance to efficiently provide low-dimensional data presentations, which could be provided by BD unsupervised techniques or auto encoders. Besides directly connecting and monitoring SDN infrastructures via OpenFlow, other techniques such as Simple Network Management Protocol (SNMP) or sFlow could be used.

However, as video traffic might be encrypted, such an approach demands for models that are capable of estimating values of the respective QoS/QoE metrics solely based on the monitored traffic parameters [34]. While these models are currently derived based on tests with human subjects, a future BD-based network optimization might even incorporate automatic QoS/QoE model creation and user inquiry. Furthermore, end-user information about the service usage could be obtained from test volunteers, who use client-side monitoring solutions or even provide feedback on QoS/QoE directly. These kind of solutions would result in a massive amount of data, which demands the efficient processing in a BD infrastructure. Such models would be updated on run-time and used by the SDN controllers.

## 1.5 Vision: Intelligent Network-Wide Auto-Optimization

With millions of transactions and events happening per second in an operator's network, the goal of leveraging this information for the purpose of quality optimization and efficiency is truly a Big Data application. Through the scalability of the cloud and new developments in analytics, it is feasible for the first time to handle this vast amount of information and gain insight into the global network state on the fly. The global network state is an accumulation of the entire network information at any point of a defined period of time. In particular recent trends in BD technologies, such as distributed data mining and information retrieval systems like Hadoop [42] or Spark [43], support such a distributed state collection and efficient processing. Adding distributed sites connected through SDN-based networks to form a telco cloud system enables to automatically act on those insights gained both globally and locally.

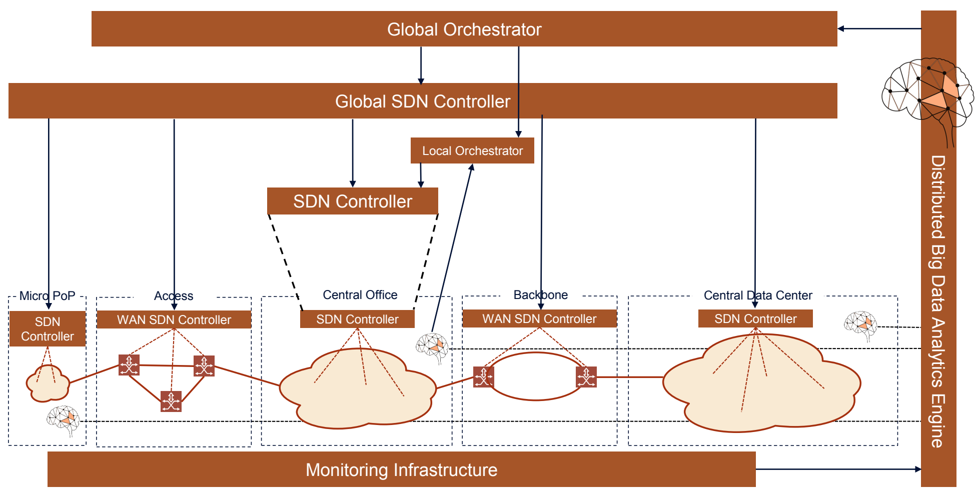



Figure 1.1: Big Data Analytics in Telco Cloud Network

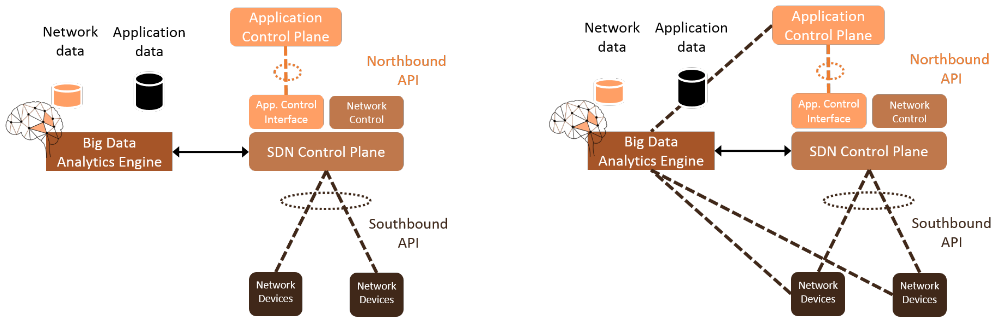
Figure 1.1 represents a high-level overview of how a Big Data Analytics Engine (BDAE) interacts with a distributed telco cloud network. An exemplary telco cloud might be structured into three tiers, namely micro points of presence (micro-pops), central offices, and central data centers. The network functionality of such telco cloud is provided by virtual network functions (VNFs) running inside the data centers of each tier. The access networks connect the micro-pops and central offices, while the backbone networks interconnect the central offices to the central data centers. The architecture follows the notion of the global-local cloud as described in the Future X Network [44]. All instances in the three tiers follow the same basic structure, consisting of computing, storage, and networking resources. The difference lies within the size and number of each category, e.g., a micro-pop might only consist of one or two servers and a small storage system connected via a small SDN network, whereas central data centers may consist of thousands of servers as well as the corresponding storage and networking equipment. Accordingly, micro-pops are large in number, while there are only a few central data centers in the network. Each of instance of every tear has its own local orchestrator as well as local SDN controller, which is shown in detail for the central office. The central office consists of several racks and is traditionally located in metro areas. For the few hyper-scale data centers, the individual data center locations are connected via high-speed optical networks.


Figure 1.2: Steps to big-data-enabled local network optimization

While this structure enables high bandwidth and flexible relocation of virtual functions between locations, a real-time optimization of user experience in a single session requires the following subsequent steps among all tiers (c.f Figure 1.2): real-time measurements, real-time analytics, real-time decisions, and real-time actions. In particular, the information exchange between tiers is important to enable a global optimization of the network operation. However, even with the increasing processing capabilities of big data applications the overhead of exchanging every piece of information would be too large. Thus, comprehensive and compact representations are needed, which could be provided by Big Data applications pre-processing the information first locally in each tier. Further, with such a concept, even latent variables of the monitored networks, i.e., the state information, could be efficiently detected via BD applications. This is enabled by the telco cloud since all functions can be performed in any of the local data centers. The local monitoring system feeds information to the local analytics engine, which generates a recommended action for the local orchestrator. The orchestrator's task is then two-fold. On the one hand, it optimizes according to the analytics results the deployment of the involved virtual functions in its domain, and on the other hand, it instructs the SDN controller to to steer the network traffic accordingly.

Apart from the local optimization in every tier, the local analytics engine also identifies and compresses information that is relevant to the network on a global scale. Which information this entails and how often it is communicated to the global analytics engine depends on the preferences and optimization goal of the global orchestrator. Both, the global orchestrator and analytics engine as well as their redudancies are located in the central data centers. Together, they optimize the whole network based on macroscopic trends and longer time-scales than the local measures. By pre-processing and -selecting the information at the local sites, the network is not congested with monitoring data and the global engine only has to deal with actually relevant information. The proper granularity of information needed for such a global optimization system remains an open research question. However, if the right abstraction can be found, an operator can facilitate fundamental changes within the network, e.g., core network reconfiguration, and prediction of necessary changes to and failures of the hardware infrastructure, in an automated fashion. That way, the operator can minimize the operational cost as well as the error introduced by human configuration of the network.

An intrinsic challenge here is the identification of information that might be of global interest as well as the interaction of network elements, controllers, orchestrators, and the big data analytics engine. Accordingly, an intelligent and well designed information exchange between big data applications and SDN controllers among all




(a) Controller-centered interaction between the Big Data Analytics Engine and the SDN controller.

(b) Data-centered interaction between the Big Data Analytics Engine and the SDN controller.

Figure 1.3: Architectural options for an integrated Big Data/SDN architecture.

domains is needed. The ideal interfaces and interactions are still an open question and may vary between different scenarios. The following section will discuss two possibilities for the access network domain.

## 1.6 Challenges of a Big Data supported SDN Architecture for Enhancing Application Quality

Figure 1.3a outlines the interaction between SDN and Big Data as presented in [19]. All available monitoring data on application and network level are gathered by the SDN control plane and forwarded to a remote analytics engine. If necessary, additional flow rules can be added to the data plane to gather specific monitoring information on demand. Based on this monitoring data, stream processing approaches as outlined in 1.3 may be applied to deduce context information or control instructions. These are then passed to the control plane and can be used to enhance the application quality. Additionally, control actions may be reported back to the BDAE and used for updating QoS/QoE models, e.g., using reinforcement learning. In this scenario, the SDN control plane may constitute a bottleneck resulting in a limited number of monitoring information and control actions being forwarded to the Big Data Analytics Engine.

A less controller-centric solution featuring the interaction between the BDAE and the SDN controller is highlighted in Figure 1.3b. Apart from the monitoring data provided by the SDN controller, the BDAE is able to collect more data from an additional monitoring system or from the network elements using management protocols like SNMP, NetFlow or sFlow. Further, the BDAE might also be connected to the application control plane enabling a direct access to monitoring data of corresponding applications. This might result in the availability of more fine-grained application data and thus a more accurate view on the applications using stream processing techniques. Additionally, reinforced learning approaches might be used to enhance the QoS/QoE models based on the impact of control actions on the application quality. Nevertheless, the total amount of exchanged data might be limited due to capacity limitations between the network devices and the Big Data Analytics Engine.

To overcome such capacity restrictions the network devices require additional knowledge to forward only selected features and examples need for the analysis in the BDAE. This can be facilitated using ML models on switch and controller level just for the special task of selecting the right features and examples. These models can be learned using BD approaches in learning clusters based on the gathered monitoring information.

## 1.7 Conclusion

Today's networks are facing a larger variety of more demanding services than ever before. Applications range from high-bandwidth multimedia applications and large numbers of Iot services to low-latency industrial applications. These diverse demands combined with an increasing number of users call for a more efficient network resource control. For this purpose, context information of applications, as well as a networking architecture capable to enforce resource control actions are required. Due to the necessity to correlate large amounts of network and application-based monitoring data, big data approaches are promising solutions to derive such context information, e.g., by deducing QoS/QoE mappings and to update them based on actual data. This information can be used by an SDN controller in an operator's network to enhance the application quality for specific users.

This book chapter is a step towards a better understanding on how Big Data approaches and Big Data Analytics can be used together with SDN architectures to enhance the overall application quality. Therefore, we introduced several SDN-based approaches that aim at enhancing user-QoE by monitoring QoE-IFs and performing appropriate control actions in applications or the network. Subsequently, we highlighted the potential of Big Data Analytics to support QoS/QoE management by introducing several works that exploit ML techniques in the context of QoE. We extended this view towards a vision on how networks can optimize themselves in the future facilitated by Big Data and Machine Learning approaches. Finally, we focused on challenges and open questions on an SDN architecture that leverages Big Data for improving the application quality.

# Bibliography


[1] Qualinet White Paper on Definitions of Quality of Experience (2012); 2013. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Patrick Le Callet, Sebastian Möller and Andrew Perkis, eds., Lausanne, Switzerland, version 1.2.

[2] Varela M, Skorin-Kapov L, Moor KD, Reichl P. QoE - Defining a User-centric Concept for Service Quality. In: Chen CW, Chatzimisios P, Dagiuklas T, Atzori L, editors. Multimedia Quality of Experience (QoE): Current Status and Future Requirements. John Wiley & Sons, Ltd; 2015. p. 5–27.

[3] Kreutz D, Ramos FMV, Verssimo PE, Rothenberg CE, Azodolmolky S, Uhlig S. Software-Defined Networking: A Comprehensive Survey. Proceedings of the IEEE. 2015;103(1):14–76.

[4] Schatz R, Fiedler M, Skorin-Kapov L. In: Möller S, Raake A, editors. QoE-Based Network and Application Management. Quality of Experience: Advanced Concepts, Applications and Methods ed. Springer International Publishing; 2014. p. 411–426.

[5] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, et al. OpenFlow: Enabling Innovation in Campus Networks. SIGCOMM Comput Commun Rev. 2008;38(2):69–74.

[6] Gudivada VN, Baeza-Yates R, Raghavan VV. Big Data: Promises and Problems. Computer. 2015;48(3):20–23.

[7] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2015-2020 White Paper. Cisco Systems, Inc., San Jose, USA; 2016.

[8] Jarschel M, Wamser F, Höhn T, Zinner T, Tran-Gia P. SDN-based Application-Aware Networking on the Example of YouTube Video Streaming. In: Proceedings of the 2nd European Workshop on Software Defined Networks (EWSDN 2013). Berlin, Germany; 2013. p. 87–92.

[9] Dobrijevic O, Santl M, Matijasevic M. Ant colony optimization for QoE-centric flow routing in software-defined networks. In: Network and Service Management (CNSM), 2015 11th International Conference on. IEEE; 2015. p. 274–278.

[10] Zhu J, Vannithamby R, Rödbro C, Chen M, Andersen SV. Improving QoE for Skype video call in mobile broadband network. In: Global Communications Conference (GLOBECOM), 2012 IEEE. IEEE; 2012. p. 1938–1943.

[11] Staehle B, Hirth M, Pries R, Wamser F, Staehle D. YoMo: A YouTube application comfort monitoring tool. New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications, Tampere, Finland. 2010;p. 1–3.

[12] Petrangeli S, Wauters T, Huysegems R, Bostoen T, De Turck F. Network-based dynamic prioritization of http adaptive streams to avoid video freezes. In: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on. IEEE; 2015. p. 1242–1248.

[13] Houdaille R, Gouache S. Shaping HTTP adaptive streams for a better user experience. In: Proceedings of the 3rd Multimedia Systems Conference. ACM; 2012. p. 1–9.

[14] Bentaleb A, Begen AC, Zimmermann R. SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking. In: Proceedings of the 2016 ACM on Multimedia Conference. ACM; 2016. p. 1296–1305.

[15] Nam H, Kim KH, Kim JY, Schulzrinne H. Towards QoE-aware video streaming using SDN. In: Global Communications Conference (GLOBECOM), 2014 IEEE. IEEE; 2014. p. 1317–1322.

[16] Cofano G, De Cicco L, Zinner T, Nguyen-Ngoc A, Tran-Gia P, Mascolo S. Design and experimental evaluation of network-assisted strategies for HTTP adaptive streaming. In: Proceedings of the 7th International Conference on Multimedia Systems. ACM; 2016. p. 3.

[17] Joseph V, de Veciana G. NOVA: QoE-driven optimization of DASH-based video delivery in networks. In: INFOCOM, 2014 Proceedings IEEE. IEEE; 2014. p. 82–90.

[18] Bahga A, Madisetti V. Big data science & analytics : a hands-on approach; 2016. Available from: http://www.worldcat.org/title/big-data-science-analytics-a-hands-on-approach/oclc/953867348&referer=brief_results.

[19] Cui L, Yu FR, Yan Q. When big data meets software-defined networking: SDN for big data and big data for SDN. IEEE Network. 2016;30(1):58–65.

[20] Mitchell T. Machine Learning. McGraw Hill; 1997.

[21] Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Springer; 2001.

[22] Murphy KP. Machine learning : a probabilistic perspective. Cambridge, Mass. [u.a.]: MIT Press; 2013. Available from: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

[23] Han J, Kamber M. Data mining: concepts and techniques. Morgan Kaufmann; 2006. Available from: http://scholar.google.de/scholar.bib?q=info:kYdwviD3IR4J:scholar.google.com/&output=citation&hl=de&as_sdt=0&scfhb=1&ct=citation&cd=0.

[24] Witten IH, Frank E, Hall MA, Pal CJ. Data mining : practical machine learning tools and techniques. Amsterdam; London: Morgan Kaufmann; 2017. Available from: http://www.worldcat.org/title/data-mining-practical-machine-learning-tools-and-techniques/oclc/972378340&referer=brief_results.

[25] Maimon O, Rokach L. Data mining and knowledge discovery handbook. vol. 14. Springer; 2010. Available from: http://scholar.google.de/scholar.bib?q=info:wwQ2svFQuQEJ:scholar.google.com/&output=citation&hl=de&as_sdt=0,5&ct=citation&cd=0.

[26] Berkhin P. Survey Of Clustering Data Mining Techniques. San Jose, CA: Accrue Software; 2002. Available from: http://citeseer.ist.psu.edu/berkhin02survey.html.

[27] Gama J. Knowledge discovery from data streams. Boca Raton, Fl.: CRC Press; 2010. Available from: http://www.worldcat.org/title/knowledge-discovery-from-data-streams/oclc/698165441&referer=brief_results.

[28] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of artificial intelligence research. 1996;4:237–285.

[29] Guo X, Singh S, Lee H, Lewis RL, Wang X. Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, editors. Advances in Neural Information Processing Systems 27. Curran Associates, Inc.; 2014. p. 3338–3346. Available from: http://papers.nips.cc/paper/5421-deep-learning-for-real-time-atari-game-play-using-offl pdf.

[30] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. Communications of the ACM. 2008;51(1):107–113.

[31] Mestres A, Rodriguez-Natal A, Carner J, Barlet-Ros P, Alarcón E, Solé M, et al. Knowledge-Defined Networking. arXiv preprint arXiv:160606222. 2016;.

[32] Clark DD, Partridge C, Ramming JC, Wroclawski JT. A knowledge plane for the internet. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM; 2003. p. 3–10.

[33] Mushtaq MS, Augustin B, Mellouk A. Empirical study based on machine learning approach to assess the QoS/QoE correlation. In: Networks and Optical Communications (NOC), 2012 17th European Conference on. IEEE; 2012. p. 1–7.

[34] Orsolic I, Pevec D, Suznjevic M, Skorin-Kapov L. YouTube QoE Estimation Based on the Analysis of Encrypted Network Traffic Using Machine Learning. In: Globecom Workshops (GC Wkshps), 2016 IEEE. IEEE; 2016. p. 1–6.

[35] Wang P, Lin SC, Luo M. A framework for QoS-aware traffic classification using semi-supervised machine learning in SDNs. In: Services Computing (SCC), 2016 IEEE International Conference on. IEEE; 2016. p. 760–765.

[36] Amaral P, Dinis J, Pinto P, Bernardo L, Tavares J, Mamede HS. Machine Learning in Software Defined Networks: Data collection and traffic classification. In: Network Protocols (ICNP), 2016 IEEE 24th International Conference on. IEEE; 2016. p. 1–5.

[37] Elkotob M, Grandlund D, Andersson K, Ahlund C. Multimedia QoE optimized management using prediction and statistical learning. In: Local Computer Networks (LCN), 2010 IEEE 35th Conference on. IEEE; 2010. p. 324–327.

[38] Du H, Guo C, Liu Y, Liu Y. Research on relationship between QoE and QoS based on BP Neural Network. In: Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on. IEEE; 2009. p. 312–315.

[39] Machado VA, Silva CN, Oliveira RS, Melo AM, Silva M, Francês CR, et al. A new proposal to provide estimation of QoS and QoE over WiMAX networks: An approach based on computational intelligence and discrete-event simulation. In: Communications (LATINCOM), 2011 IEEE Latin-American Conference on. IEEE; 2011. p. 1–6.

[40] Borchert K, Hirth M, Zinner T, Mocanu DC. Correlating QoE and Technical Parameters of an SAP System in an Enterprise Environment. In: Teletraffic Congress (ITC 28), 2016 28th International. vol. 3. IEEE; 2016. p. 34–36.

[41] It U. ITU-T Recommendation G.114. International Telecommunication Union; 1993.

[42] Apache Software Foundation. Apache Hadoop;. Available from: `http://hadoop.apache.org/`.

[43] Apache Software Foundation. Apache Spark: Lightning-fast cluster computing;. Available from: `http://spark.apache.org/`.

[44] Weldon MK. The future X network: a Bell Labs perspective. Crc Press; 2016.