

RESEARCH

Open Access



A quadri-dimensional approach for poor performance prioritization in mobile networks using Big Data

Maluambanzila Minerve Mampaka*  and Mbuyu Sumbwanyambe

*Correspondence:
minervemm@gmail.com
Department of Electrical
and Mining Engineering,
University of South Africa,
Johannesburg 1710, South
Africa

Abstract

The Management of mobile networks has become so complex due to a huge number of devices, technologies and services involved. Network optimization and incidents management in mobile networks determine the level of the quality of service provided by the communication service providers (CSPs). Generally, the down time of a system and the time taken to repair [mean time to repair (MTTR)] has a direct impact on the revenue, especially on the operational expenditure (OPEX). A fast root cause analysis (RCA) mechanism is therefore crucial to improve the efficiency of the operational team within the CSPs. This paper proposes a quadri-dimensional approach (i.e. services, subscribers, handsets and cells) to build a service quality management (SQM) tree in a Big Data platform. This is meant to speed up the root cause analysis and prioritize the elements impacting the performance of the network. Two algorithms have been proposed; the first one, to normalize the performance indicators and the second one to build the SQM tree by aggregating the performance indicators for different dimensions to allow ranking and detection of tree paths with the worst performance. Additionally, the proposed approach will allow CSPs to detect the mobile network dimensions causing network issues in a faster way and protect their revenue while improving the quality of the service delivered.

Keywords: Big Data, QoS, QoE, MTTR, Root cause analysis, SQM, CEM, Mobile networks

Introduction

Communication service providers (CSPs) have reached the ceiling in terms of new customer acquisitions. Therefore, acquiring new customers is much difficult than it is for existing customers to churn. Traditional network operation centres (NOC) have been very inefficient in terms of problem finding, handling and resolution. Within this ambit, and driven by the need for fast service, the NOC approach of managing network incidents has changed to the new paradigm of service quality management (SQM) and customer experience management (CEM). This requires mobile network operators to be more service-oriented and customer-oriented by using the service operation centre (SOC) approach.

While the traditional approach of mobile network monitoring follows a bottom-up approach, i.e. Starting with the network elements management, network alarming and quality of service (QoS) issues through historical key performance indicators (KPI)

monitoring; the SOC, through the SQM/CEM, follows the top-down approach starting with the aggregated service quality index (SQI) and down to the KPIs. The SOC does not only assist the CSPs to have a faster reaction to issues, but also, to act proactively based on statistical values and forecast [1].

Some of the benefits of an SQM/CEM are in the reduction of operational expenditure (OPEX), reduction of time-to-market, mean time to repair (MTTR) and increase in the revenues. However, there are still barriers regarding the full implementation of the SQM/CEM. The barriers relate to the lack of skills and experience in implementing the SQM/CEM and to some extent the complexity that comes with handling Big Data required in the extraction of essential values from the network. The correlation between different parts of the network is a necessity during the design process of the SQM/CEM. The necessity arises because of the need to provide an end-to-end quality of experience (QoE) for customers across different interfaces and touch points across the network. The customer-centric metrics can be computed only by collecting information from different parts of the network [2]. These metrics are weighted functions of the aggregated network SQI attributes and KPIs from different services and different levels in the network.

To ensure a proper QoS and QoE, the CSPs usually consider multiple data sources that may include the call data records (CDRs), the measurement reports and the operations and maintenance centre (OMC) data to provide insights into different areas of the business organization [3]. With multiple data sources and correlation requirement, the CSPs need to manage intelligently large amount of data. This is done through the usage of technologies such as Big Data analytics and machine learning [4]. Generally, when diagnosing a network or fixing any network problem, CSPs spend a lot of time in doing so because of the complexity of services and the number of elements involved in the mobile networks. To evaluate the efficiency in detecting and solving issues in the network, the CSPs make use of the MTTR [5].

In this paper, a computational approach is proposed to enhance and manage the QoS based on the four mobile network dimensions. These dimensions are the service, subscriber, the handset and the network element (The cell in this case). The proposed approach builds a performance quality weighted-tree to enable root cause analysis (RCA) by determining, in a faster way, the worst paths in the tree. This approach reduces the MTTR since it provides a fast way of finding network issues.

This paper discusses some concepts of Big Data and the RCA for the telecom industry in “[Background and related works](#)” section. “[Methods](#)” section describes the system architecture and the proposed approach used to design the SQM. “[Results and discussion](#)” section provides the output of the research. The last, “[Conclusions](#)” section summarizes the research.

Background and related works

The key concepts about the applications of Big Data and the RCA in the telecommunication networks are discussed in this section.

Big Data and Hadoop

The term Big Data is often used for a large amount of data that is difficult to manage using the traditional database management tools. Big Data in telecommunication

requires fast processing and scalable platforms [6]. Some of the platforms that are used in handling Big Data are oracle DB2, EMC Greenplum, Vertica, Microsoft PDW, Teradata and Hadoop [7]. Hadoop is an open-source software platform which is implemented using the Java language. It allows the storage of large files on a single machine or in a cluster of computers for distributed processing of huge datasets. The main components of the Hadoop ecosystem are the Hadoop distributed file system (HDFS) and the MapReduce framework. The HDFS manages the storage of large files while the MapReduce framework distributes the tasks across several nodes by processing the input data and producing intermediate results in the Map phase at the same time merging the intermediate results that have the same keys in the Reduce phase [8].

Big Data in telecommunication

Big Data plays an important role in the telecommunication industry. Its impact is visible in key managerial decisions and organizational practices that contribute to the CSPs' profit [9]. Several studies have been conducted for the usage of Big Data in the mobile network environment. He et al. [4] proposed a unified data model for an architectural framework based on the random matrix theory and application of machine learning techniques for Big Data analytics in the mobile networks. The paper also illustrated examples of Big Data application in mobile network such as for data traffic, location, signalling, heterogeneous networks and radio waveforms. The research concluded with open research challenges in Big Data application in mobile networks such as data privacy, filtering and compression. Extant literature such as Su et al. [3], proposed a Big Data platform to collect, process and analyse the large amounts of data in telecommunication networks. A Hadoop-based and a multiple parallel processing database architecture was applied on network data. The approach was used to achieve a unified management and storage system with the capability to diagnose the problems and optimize the network. The results of this study demonstrated a better performance in Big Data loading, storage and analysis compared to the traditional data warehousing, showing the benefits of using Big Data technologies.

To enable CSPs to manage the network resources in an effective and efficient way and support better QoS, Si et al. [8] developed a Big Data analysis platform to analyse the mobile network traffic data patterns for the resource management and the usage of the network elements. Two datasets were used in Apache Hadoop for the storage and Mahout for the implementation of machine learning algorithms. The platform incorporated the K-means and fuzzy K-means for clustering. The results of the paper focused on improving the execution time by changing the Hadoop cluster parameters and making use of the pre-processing data as well as machine learning techniques. Jun et al. [10] collected CSPs core network data and proposed Zipf-like models to analyse and cluster traffic distribution volumes between service providers and the subscribers. The model essentially solved a time series unsupervised clustering challenge by identifying the traffic patterns. The results of this study highlighted the users' behaviours leading to the traffic patterns and the service categories used.

Çelebiet al. [7] used a Big Data approach to analyse handovers from 3G to 2G mobile networks. The study proposed an analysis of the A interface signalling messages between the base station subsystem (BSS) and the mobile switching centre (MSC). Due to the

large amount of signalling messages, a Hadoop platform was used to load the data into the HDFS. The queries were executed using Apache Hive to transform structured query language (SQL) into MapReduce functions. The results provided an insight into 3G service holes (areas with service discontinuity). This outperformed the base station KPIs analysis approach in terms of accuracy.

Jie et al. [11] used a distributed computing Hadoop based system to analyse high-speed network traffic from massive data captured on a 3G network. The internet traffic from smartphones was analysed to leverage a MapReduce parallel programming model with the objective of understanding the usage patterns and the forecast growths of the network traffic. The data was collected using a traffic monitoring system deployed at the Gn interface between the serving general packet radio service (GPRS) support node (SGSN) and the gateway GPRS support node (GGSN). The results of this research provided flow characteristics of smartphone operating systems and their related traffic which could be useful for CSPs to anticipate the fast traffic growth in the network.

Root cause analysis (RCA) in telecommunication

The full automation of processes in telecommunication network management will still take time and therefore the support of human expertise is still needed. The evolution of the technologies and the proliferation of handsets and services creating huge numbers of errors and faults have increased the scale of complexity for the incident management and the RCA.

In line with the above, Botta et al. [12] proposed an intelligent customer service assurance platform for mobile broadband network. To support advanced operation support system (OSS), a multidimensional root cause analysis was done on the network architecture with a view of improving the bit rate, and to some degree correlate the control and the user plane. The result of the research was used on a real network to provide benefits for the mobility and the session management as well as the transmission control protocol (TCP) connections and enhance the RCA.

Keeney et al. [13] designed and recommended a system that was to be used in assisting the NOC operational team to manage incidents occurring in the network. The approach consisted of a collection of telecommunication data from the OSS in an intelligent way. The data was then analysed and correlated so as to provide prediction for proactive maintenance.

Parwez et al. [14] proposed an intelligent model to predict anomalies in the network through Big Data and machine learning algorithms. The model used a hierarchical clustering approach and a neural network model to analyse the network traffic from spatio-temporal call detail records. Vega et al. [15] proposed a time series analysis for anomaly detection with a proactive and reactive approaches. The proactive approach was based on the behavioural analysis of the historical data and the reactive one, based on traffic disruption on the time series data. To determine the root causes of network performance issues, statistical thresholds of performance metrics were used and correlated with each other. Wang and Handurukande [16] used a similar time series approach and provided principles to design a network management system in the context of network functions virtualization (NFV) and software defined networking (SDN). The stream

analytics engine proposed used the presence of abnormalities within the network counters and KPIs to identify network problems.

The results from [12–16] were essentially from time series analysis and focused on the network elements without providing the impact of the network issues on the subscribers or the influence of the handsets on the overall network performance.

Kingsley and Dahj [17] proposed a tree-based SQM approach for efficient low-cost service management with a particular focus on the over-the-top (OTT) applications. The SQM-tree had four levels that focused on the 3G mobile networks services classes, i.e. Streaming, interactive and down to the OTT applications. The system connected to a cloud application so as to provide reporting and Big Data throughput. SparkSQL was used to query the stored data allowing a drilldown to worst cellular cells, devices and subscribers. One of the drawbacks of that proposed system was the class of service classification which was specific to 3G mobile networks and should be reviewed for a different technology. Two other methods, still focusing on the OTT internet services, were proposed by Fiadinoet al [18], resulting in the development of a framework called RCATool. The RCATool used the domain name server (DNS) protocol to detect and diagnose the traffic anomalies in the network. Diagnostic features such as devices information, error codes and the hostnames were used in the investigation of problems. The first method applied to the entropy of the diagnostic features while the second method considered the statistical distribution of features such as the traffic. Miyazawa and Nishimura [19] proposed an RCA approach in investigating service failures in a fixed-mobile converged network. The approach used alarms classification and a hierarchical alarm data model on different types of alarms such as the resource alarms, the performance alarms and the service alarms to pinpoint the cause of network failures. In essence [18, 19], analysed only one type of services, which is far from the current reality of mobile networks implementation. Cai et al. [20] provided an intensive investigation of fault diagnosis using Bayesian network. Although Bayesian network works well for fault diagnosis even for complex industrial systems, it has drawbacks for non-permanent faults that provides only weak signals and for online fault diagnosis which are very slow. In line with the above, Hong et al. [21] investigated fault diagnosis for the circuit-switched fall-back service. Although the investigation used detailed data of the signalling procedures from different mobile operators, the problem finding mechanism was manual and not subscriber-oriented.

Pablo et al. [22] proposed a self-healing algorithm in the context of self-organized network (SON) to increase the CSPs' revenues. The algorithm was based on a temporal evaluation of the network metrics to detect and diagnosis the issues. In order to reduce the diagnosis error rate, correlations between different metrics from the radio access network were investigated. The proposed algorithm was compared with a fuzzy logic approach providing different results depending on the network elements involved. Palacios et al. [23] and Hahn et al. [24] also proposed methods based on SON. The authors in [23] proposed an automatic selection of KPIs algorithm based on the overlapping area of the probability density function. This allowed analysis of statistical behaviour of the network states and performance indicators. The results from the method outperformed troubleshooting expert ones. The authors in [24] focused on multi-radio access technologies handover investigation. Different KPIs for mobility and traffic steering were

analysed to provide classification of network cells based on network load and the handover success rate. The patterns of different cell classes were used to enhance the performance of a SON system as a function for a future root cause analysis. Although some of the network related issues can be analysed and automatically solved by the SON, most of the SON systems are not fully autonomous and still rely on experts' validation.

Lastly, Laselva et al. [25] proposed a framework to assess the QoE built from an aggregation model using the network and service KPIs. The approach considered three network dimensions which are the subscribers from the QoE, the service and the network elements based on the KPIs. The device dimension was not considered and there was no splitting of KPIs per services.

Methods

The proposed approach has the objective of improving the RCA in mobile networks, driven by Big Data, and reducing the complexity of the optimization of the services and other network related elements. This paper proposes a system model following an SQM-tree approach with nodes. The information held in the node was used as a path for sorting and prioritization of tree paths. This was done in order to understand which network dimensions and KPIs are influencing the performance of the network negatively.

System architecture

The system architecture consisted of a physical laptop (Computer1) running R through RStudio [26] and a virtual machine (VM) based on VMware [27] running a single node Cloudera platform with Hadoop as shown in Fig. 1.

Computer1 was used to prepare scripts and queries using R programming language, and to connect via a Cloudera Impala connector to the VM which contained the HDFS where the SQM file was stored. The Cloudera Impala [28] is a massively parallel processing (MPP) SQL query engine for Apache Hadoop. The configuration of the hardware used to conduct the experiment are shown in Table 1.

The SQM file stored in the HDFS had eleven columns with one million records aggregated based on four keys and seven core network performance indicators. The four keys were the service, the subscriber based on the international mobile subscriber identity (IMSI), the handset based on the type approval code (TAC) and the cell based on the cell-ID (A unique Identifier of a cell in the network). The seven core network

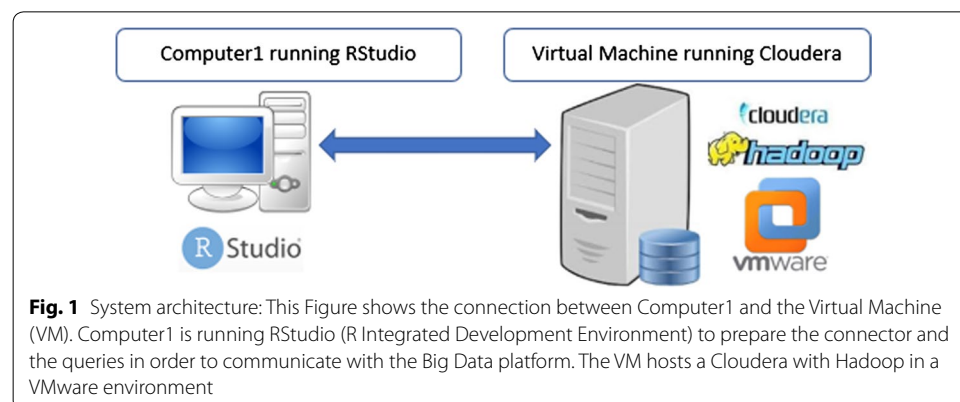


Table 1 Hardware configuration

Parameter	Value
Computer1 Processor	Intel Core i7 (4 Processors)
Computer1 Memory	16 GB (8 GB used for the VM)
Computer1 storage	1 Terabytes HDD
Computer1 IP	198.168.68.1
VM processor	Hosted on VMware in Computer1 (4 processors)
VM memory	8 GB
VM storage	64 GB HDD
VM IP	198.168.68.128

Table 2 SQM fields details

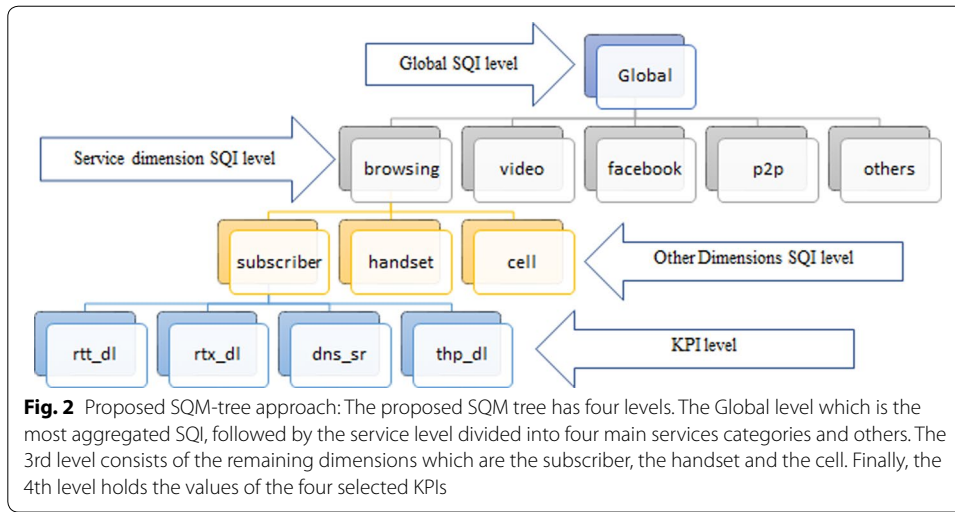
Field name	Description	Example
Service	The types of services	Browsing
Subscriber	The subscriber ID (based on the IMSI)	*****13992
Handset	The device type (based on the TAC)	35195507
Cell	The cell-ID	*****467
Events	The number of events for every aggregation	3
sec_dl	The active second on the downlink	148
retransbytes_dl	The bytes retransmission on the downlink	655594
bytes_dl	The bytes transmitted on the downlink	2451800
dns_successful	The number of success full DNS transaction	80
dns_failure	The number of failed full DNS transaction	0
latency_dl	The latency on the downlink	518

performance indicators were the total number of events (events), the total time of data connection (sec_dl), the total bytes retransmitted on the downlink (retransbytes_dl), the total bytes transmitted on the downlink (bytes_dl), the number of successful DNS transactions (dns_successful), the number of unsuccessful DNS transactions (dns_failure) and the latency from the core to the user equipment (latency_dl). The details of the SQM file are shown in Table 2.

Quadri-dimensional approach

To implement the quadri-dimensional approach, an SQM-tree was built based on four dimensions which were the service, the subscriber, the handset and the cell with four levels representing the depth of the tree nodes as shown in Fig. 2.

From the tree shown in Fig. 2, the global level consisted of the highest aggregation of the whole network performance. Down from the global level, the first SQI dimension level was computed. This, consisted of services such as browsing, video, facebook, peer-to-peer (p2p) and others (Representing the rest of the traffic categories). The second SQI dimension level consisted of the subscriber, the handset and the cell. The third one was the KPI level which consisted of the round-trip time on the downlink (rtt_dl), the retransmission rate on the downlink (rtx_dl), the DNS success rate (dns_sr) and the throughput on the downlink (thp_dl).



To make sure only meaningful data transactions were considered, the approach used a flag for transactions with bytes transmitted on the downlink above 1.5 Megabytes and an active connection time above 50 s. Only the records with the flag set were considered. The formulae used to calculate the KPIs are as follows:

$$thp_dl = \begin{cases} \frac{8 * (\sum bytes_dl)}{1024 * (\sum sec_dl)}, & flag > 0 \\ N/A, & flag \leq 0 \end{cases} \quad (1)$$

$$rtx_dl = \frac{100 * (\sum retransbytes_dl)}{\sum bytes_dl} \quad (2)$$

$$dns_sr = \frac{100 * (\sum dns_successful)}{\sum (dns_successful + dns_failure)} \quad (3)$$

$$rtt_dl = \frac{\sum latency_dl}{\sum events} \quad (4)$$

SQM-tree construction

To build the SQM-tree, two algorithms were used. The first one was used to normalize the KPI level and the second one was used to build and fill in the SQM tree following the quadri-dimensional approach.

Since the KPIs such as the throughput (thp_dl) and the round-trip time (rtt_dl) are numbers that can range from zero to several thousands, the first algorithm (algorithm 1) as shown in Fig. 3, was used to normalize the KPI level by receiving the original KPI value Kk and returning a normalized value Kk' ranging between 0 and 100. Since the data used were based on a 3G packet-switched mobile network, thp_dl values less than 500 Kbps were considered as worst and normalized to 0, values ranging from 500 Kbps

Algorithm 1 KPI level normalization

```

1:      Input: Kk
2:      Output: Kk'
3:      if Kk == 'thp_dl' then
4:          if Kk > 1024 then
5:              Kk' = 100;
6:          else if Kk > 500 then
7:              Kk' = 100 * (Kk - 500) / 500;
8:          else
9:              Kk' = 0;
10:         end if
11:     else if Kk == 'rtt_dl' then
12:         if Kk < 500 then
13:             Kk' = 100;
14:         else if Kk < 1000 then
15:             Kk' = 100 * (1000 - Kk) / Kk;
16:         else
17:             Kk' = 0;
18:     else if Kk == 'rtx_dl' then
19:         Kk' = 100 - Kk;
20:     else
21:         Kk' = Kk
22:     end if

```

Fig. 3 Algorithm 1 for KPI level normalization: To normalize all the values of KPIs to be between 0 and 100, Algorithm 1 sets the lower and the upper boundaries of each KPIs depending of the threshold for the best and worst values. This was applied for the throughput downlink (thp_dl) in Kbps, the round-trip time on the downlink (rtt_dl) in milliseconds and the retransmission on the downlink (rtx_dl) in percentage

to 1 Mbps were normalized between 0 and 100 while values above 1 Mbps were considered as best and normalized at 100. For the rtt_dl, values less than 500 ms were considered as best and normalized at 100, values between 500 and 1000 ms were normalized between 100 and 0 while values above 1000 ms were considered as worst and normalized at 0. For the rtx_dl, as it was in percentage with 0% as the best value, the normalized value was considered as the complementary in respect to 100. Finally, the dns_sr remained the same as it was already a percentage and 100% was the best value.

Algorithm2 SQM-tree Construction

```

1: Input: D, S, K
2: Output: SQM-tree
3: initialise global = node. New("global");
4: for each Si in S do
5:   serv = global.AddChild(Si);
6:   for each Dj in D do
7:     if Dj != "service" then
8:       dim = serv.AddChild(Dj);
9:       for each Kk in K do
10:        kpi = dim.AddChild(Kk);
11:        query1 = "SELECT Algorithm1(Kk) FROM SQM WHERE service = Si";
12:        query2 = "SELECT COUNT(DISTINCT(Dj)) FROM SQM WHERE service = Si AND Algorithm1(Kk)>50";
13:        query3 = "SELECT COUNT(DISTINCT(Dj)) FROM SQM WHERE service = Si";
14:        global.Si.Dj.Kk.value= dbGetQuery(impala,query1);
15:        global.Si.Dj.Kk.impact= 100*dbGetQuery(impala,query2)/dbGetQuery(impala,query3);
16:        global.Si.Dj.Kk.quality = 0.5*global.Si.Dj.Kk.value + 0.5*global.Si.Dj.Kk.impact;
17:      end for
18:      global.Si.Dj.value = 0.25*global.Si.Dj.K1.value + 0.25*global.Si.Dj.K2.value +
        0.25*global.Si.Dj.K3.value+ 0.25*global.Si.Dj.K4.value;
19:      global.Si.Dj.impact = 0.25*global.Si.Dj.K1.impact + 0.25*global.Si.Dj.K2.impact +
        0.25*global.Si.Dj.K3.impact + 0.25*global.Si.Dj.K4.impact;
20:      global.Si.Dj.quality = 0.5*global.Si.Dj.value + 0.5*global.Si.Dj.impact;
21:    end if
22:    global.Si.value = 0.5*global.Si.D2.value + 0.25*global.Si.D3.value+ 0.25*global.Si.D4.value;
23:    global.Si.impact = 0.5*global.Si.D2.impact + 0.25*global.Si.D3.impact + 0.25*global.Si.D4.impact;
24:    global.Si.quality = 0.5*global.Si.value + 0.5*global.Si.impact;
25:  end for
26: end for
27: global.value = 0.2*global.S1.value + 0.2*global.S2.value + 0.2*global.S3.value +
  0.2*global.S4.value + 0.2*global.S5.value;
28: global.impact = 0.2*global.S1.impact + 0.2*global.S2.impact +
  0.2*global.S3.impact + 0.2*global.S4.impact + 0.2*global.S5.impact;
29: global.quality = 0.5*global.value + 0.5*global.impact;

```

Fig. 4 Algorithm 2 for SQM-tree construction: the algorithm 2 dynamically generate the queries essential to construct and fill in the SQM tree. The queries are launched using an impala connector to the Big Data platform and the results of the KPIs are combined with some weights (considering the subscribers as the most important dimension) to derive the SQIs for each tree paths

The second algorithm (algorithm 2) as shown in Fig. 4, was the algorithm used to construct the SQM-tree nodes based on the quadri-dimensional approach focusing on the four dimensions (Service, subscriber, handset and cell) and to dynamically build the Big Data queries to fill in the tree with both the KPIs and the aggregated SQIs data. All the

Table 3 MySQL vs Cloudera performance comparison

Test	MySQL execution time (s)	Cloudera impala execution time (s)
Query1	2.2	0.7
Query2	6.1	2.3

nodes in the tree had three information which were the “value”, the “impact” and the “quality”. The additional information was added to provide quality indicators that do not only consider the aggregated values of the KPIs, but also the impact of the performance on each dimension. The “value” was the weighted aggregation of different KPIs, the “impact” was the percentage of a dimension (subscriber, handset, etc.) with better service performance (With the normalized KPIs > 50) and the “quality” was the weighted aggregation of both the “value” and the “impact”. The algorithm 2 received three sets of data and returned the built and filled SQM-tree following the quadri-dimensional approach. The sets of data received were the service set S defined as $S = \{\text{“browsing”}, \text{“video”}, \text{“facebook”}, \text{“p2p”}, \text{“others”}\}$ where S_i represented each service with $i \in \{1, 2, 3, 4, 5\}$, the dimension set D defined as $D = \{\text{“service”}, \text{“subscriber”}, \text{“handset”}, \text{“cell”}\}$ where D_j represented each dimension with $j \in \{1, 2, 3, 4\}$ and the KPI set K defined as $K = \{\text{“rtt_dl”}, \text{“rtx_dl”}, \text{“dns_sr”}, \text{“thp_dl”}\}$ where K_k represented each KPI with $k \in \{1, 2, 3, 4\}$.

Results and discussion

Cloudera vs MySQL performance comparison

The experiment was run both in a Big Data platform and in MySQL. Table 3 shows the average performance comparison between MySQL and Cloudera Impala using the queries from algorithm 2. The results show that using a Big Data platform, even running on a single machine had a performance at least three times better than the traditional MySQL.

SQM-tree output

The SQM-tree output of the algorithm 2, had the “value”, the “impact” and the “quality”. All of them ranged from 0 to 100 for the Global, SQI and the KPIs levels represented by the three leaves as “levelName”. Figure 5 shows the results of the SQM-tree where it was possible to analyse specific “levelName” ordered by the “quality” and to identify the worst paths instead of running multiple queries and KPIs analysis as currently done in most of the NOC. The benefit of the quadri-dimensional approach was in its fast troubleshooting and RCA capability.

Since the SQM-tree paths were built following the quadri-dimensional approach, the worst paths provided information about which dimensions (service, handset, subscriber, cell) and KPIs had an impact on the QoS and the QoE.

Worst SQM-tree paths based on the performance quality

Figure 6 shows a screenshot of the 10 worst paths ranked by the performance quality. From the list of the worst paths, it was possible to conclude that the worst KPI was

	levelName	value	impact	quality
1	global	75.30594	78.0472175	76.6765786
2	--browsing	73.03688	77.7868530	75.4118685
3	--subscriber	73.03688	74.4733841	73.7551341
4	--rtt_dl	100.00000	96.6186745	98.3093373
5	--rtx_dl	97.17403	99.9716362	98.5728335
6	--dns_sr	94.97351	99.0492097	97.0113577
7	°--thp_dl	0.00000	2.2540158	1.1270079
8	--handset	73.03688	78.1595347	75.5982094
9	--rtt_dl	100.00000	97.1416701	98.5708351
10	--rtx_dl	97.17403	99.9667636	98.5703972
11	--dns_sr	94.97351	99.3768176	97.1751617
12	°--thp_dl	0.00000	16.1528874	8.0764437
13	°--cell	73.03688	84.0411090	78.5389966
14	--rtt_dl	100.00000	99.9584757	99.9792379
15	--rtx_dl	97.17403	100.0000000	98.5870154
16	--dns_sr	94.97351	99.9744466	97.4739762
17	°--thp_dl	0.00000	36.2315137	18.1157569

Fig. 5 SQM-tree result screenshot: This Figure shows a partial result of the SQM-tree. Each node of the tree holds three indicators which are the results of Algorithm 2. Those indicators are the value, the impact and the quality (which is an aggregation of the value and the impact). Those three metrics are in percentage and useful by simple sorting to determine the worst SQM-tree paths

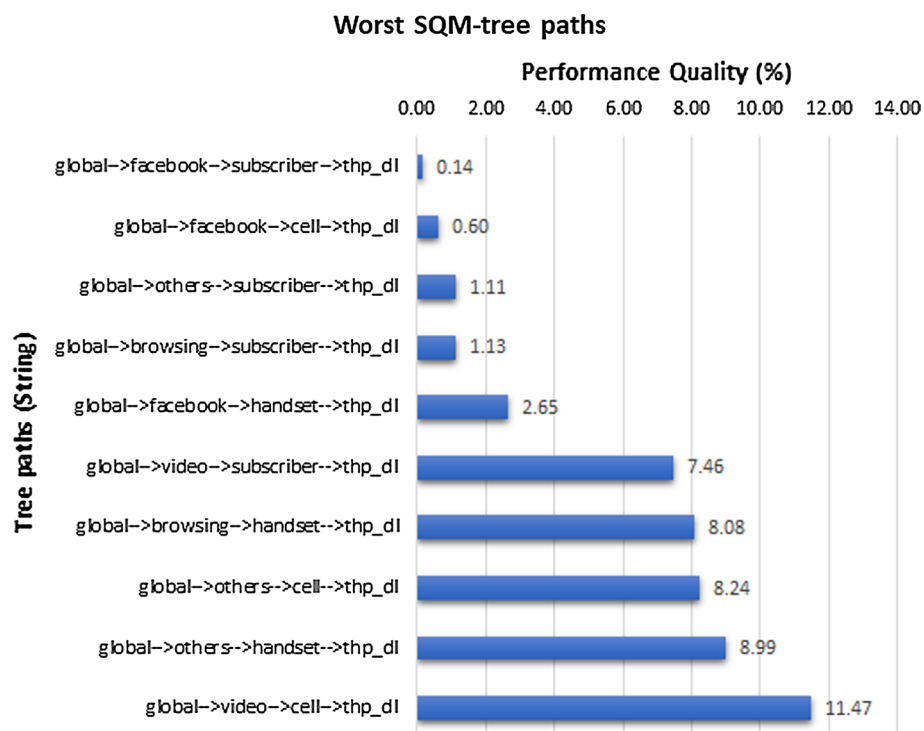


Fig. 6 Ten worst SQM-tree paths: This figure shows the worst SQM tree paths when sorting the performance quality in ascending order. The performance quality is expressed in percentage with the lower value representing the worst performance. In this example the most impacted KPI is the throughput downlink (thp_dl) which affects randomly different dimensions and services

the throughput on the downlink affecting different dimensions and services. The most impacted service was the “facebook” service with lower throughput affecting, essentially, the subscriber and the cell dimensions.

The SQM-tree results can lead to an investigation and troubleshooting of network. This can be done by prioritizing the paths with poor performances to reduce the MTTR as the time to detect the issues can be sensibly reduced. This, in essence, will improve the efficiency of the CSPs operation team.

Discussion

Several approaches have already been proposed to provide RCA for CSPs. Most of these approaches focused on either specific network dimension such as the network element or the service. This paper proposed an approach to overcome some of the RCA issues based on three arguments.

The first argument is to go beyond the time series approaches focusing only on a single network dimension as proposed in studies such as [12–16]. The SQM-tree approach proposed in this paper is based on an aggregation scheme. This SQM-tree can be constructed for different time granularities to provide the same benefits as a time series analysis while considering multiple dimensions (subscribers, device, network elements and the service) in the RCA process. Furthermore, the proposed approach allows multiple service investigation to improve the RCA capabilities and enhance single service approaches as proposed by [18, 19].

The second argument is to propose a technology-agnostic solution. The method proposed by Kingsley and Dahj [17] was built focusing on the service classes of 3G mobile networks. As mobile network technologies are evolving, different classes of services and applications are emerging. The SQM-tree proposed in this paper can be used for multiple technologies. This allow operators to use the same system for upcoming technologies such as 5G mobile network with different classes of services.

The third argument is to have a scalable and adaptive system. Unlike methods such as the one proposed by Laselva et al. [25], where only traffic analysis and alarms threshold were taken into consideration, this paper used a Big Data approach. The Big Data approach is dynamic and scalable as it consists of an aggregation scheme to simplify not only a fast RCA process but also the addition of new services or new KPIs for the future. The proposed approach takes advantage of the Big Data features such as the scalability and flexibility.

Conclusion

In this paper, an SQM design approach was proposed, considering the four dimensions involved in the mobile networks (service, subscriber, handset and the cell). The SQM designed followed a tree approach based on a KPI normalization algorithm and an SQM-tree construction algorithm. The SQM-tree construction algorithm dynamically prepared the Big Data queries essential for the tree node weights. The tree nodes held values not only based on KPI aggregation but also from the impact of the KPI on the mobile network’s dimensions. The final tree results were then sorted to provide faster RCA and prioritization in managing issues affecting the network most. A performance comparison was also done between the Big Data platform and the traditional MySQL to

demonstrate that even running on single machine, the Big Data platform can have better performance.

Abbreviations

MTTR: mean time to repair; CSPs: communication service providers; OPEX: operational expenditure; SQM: service quality management; NOC: network operation centre; CEM: customer experience management; SOC: service operation centre; QoS: quality of service; KPIs: key performance indicators; SQL: service quality index; QoE: quality of experience; CDRs: call data records; OMC: operations and maintenance centre; HDFS: Hadoop distributed file system; SQL: structured query language; SGSN: serving GPRS support node; GGSN: gateway GPRS support node; RCA: root cause analysis; OSS: operation support system; TCP: transport control protocol; OTT: over-the-top; VM: virtual machine; MPP: massively parallel processing; IMSI: international mobile subscriber identity; TAC: type approval code; SDN: software-defined networking; NFV: network function virtualization.

Authors' contributions

MMM and MS conceived and designed this study. MMM implemented the experiment. Both authors read and approved the final manuscript.

Acknowledgements

We thank Sonia Kiangala for providing assistance with the cleaning of the data.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets generated and/or analysed during the current study are not publicly available due to the visibility on the performance of the involved mobile operator. But a sample of the anonymized data is available from the corresponding author on reasonable request.

Consent for publication

The authors consent to the publication of this work.

Funding

This work was supported by the University of South Africa (UNISA).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 12 November 2018 Accepted: 18 January 2019

Published online: 04 February 2019

References

1. Banovic-Curguz N, Ilisevic D. Moving from network-centric toward customer-centric CSPs in bosnia and Herzegovina. In: 39th international convention on information and communication technology, electronics and microelectronics (MIPRO), 2016. P. 696–701.
2. Monserrat JF, Alepuz I, Cabrejas J, Osa V, López J, García R, Domenech MJ, Soler V. Towards user-centric operation in 5G networks. *EURASIP J Wireless Commun Netw*. 2016;2006:1–7.
3. Su F, Peng Y, Mao X, Cheng X, Chen W. The research of Big Data architecture on telecom industry. In: 16th international symposium on communications and information technologies (ISCIT), 2016. p. 280–4.
4. He Y, Yu FR, Zhao N, Yin H, YaoH Qiu RC. Big data analytics in mobile cellular networks. *IEEE Access*. 2016;4:1985–96.
5. Bokun S, He H, Rao A. A SOC evolves from a cost centre to a revenue centre for some CSPs. *Analysys Mason*, UK. 2016.
6. Maria V, Mone F. Big data services based on mobile data and their strategic importance. In: 2018 7th international conference on computers communications and control (ICCCC), Oradea, 2018. P. 276–81.
7. Çelebi ÖF, et al. On use of Big Data for enhancing network coverage analysis. *ICT*. 2013;2013:1–5.
8. Si M, Lung C, Ajila S, Ding W. An empirical investigation of mobile network traffic data for resource management. In: 2016 IEEE international congress on big data (BigData Congress), San Francisco, CA, 2016. P. 291–8.
9. Bughin J. Reaping the benefits of big data in telecom. *J Big Data*. 2016;3:14. <https://doi.org/10.1186/s40537-016-0048-1>.
10. Jun L, Tingting L, Gang C, Hua Y, Zhenming L. Mining and modelling the dynamic patterns of service providers in cellular data network based on Big Data analysis. *China Commun*. 2013;10(12):25–36.
11. Jie Y, Shuo Z, Xinyu Z, Jun L, Gang C. Characterizing smartphone traffic with MapReduce. In: International symposium on wireless personal multimedia communications, WPMC, 2013. P. 1–5.
12. Botta A, Pescapé A, Guerrini C, Mangri M. A customer service assurance platform for mobile broadband networks. *IEEE Commun Magazine*. 2011;49(10):101–9.
13. Keeney J, Van der Meer S, Hogan G. A recommender-system for telecommunications network management actions. In: 2013 IFIP/IEEE international symposium on integrated network management (IM), 2013. P. 760–3.

14. Parwez MS, Rawat DB, Garuba M. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Trans Ind Inf.* 2017;13(4):2058–65.
15. Vega C, Aracil J, Magana E. KISS Methodologies for Network Management and Anomaly Detection. In: 2018 26th international conference on software, telecommunications and computer networks (SoftCOM), Split, Croatia, 2018. P. 1–6.
16. Wang M, Handurukande SB. Anomaly detection for mobile network management. *Int J Next-Gener Comput.* 2018;9(2):80–97.
17. Kingsley OA, Dahj JN. Modeling of an efficient low cost, tree based data service quality management for mobile operators using in-memory big data processing and business intelligence use cases. In: 2018 international conference on advances in big data, computing and data communication systems (icABCD), At Uhlanga, Durban. South Africa. 2018. <https://doi.org/10.1109/icabcd.2018.8465410>.
18. Fiadino P, DAlconzo A, Schiavone M, Casas P. RCATool—a framework for detecting and diagnosing anomalies in cellular networks. In: 2015 27th international teletraffic congress, Ghent, 2015. P. 194–202.
19. Miyazawa M, Nishimura K. Scalable root cause analysis assisted by classified alarm information model-based algorithm. In: 2011 7th international conference on network and service management, Paris, 2011. P. 1–4.
20. Cai B, Huang L, Xie M. Bayesian networks in fault diagnosis. *IEEE Trans Industr Inf.* 2017;13(5):2227–40.
21. Hong B, et al. Peeking over the cellular walled gardens—a method for closed network diagnosis. *IEEE Trans Mob Comput.* 2018;17(10):2366–80.
22. Li Z, Ouyang Y, Su L, Jiang W, Hu Y, Lin Z. Detecting traffic anomaly in wireless networks, an analytics methodology. In: 2018 wireless telecommunications symposium (WTS), Phoenix, AZ, 2018. P. 1–6.
23. Palacios D, De-la-Bandera I, Gómez-Andrades A, Flores L, Barco R. Automatic feature selection technique for next generation self-organizing networks. *IEEE Commun Lett.* 2018;22(6):1272–5.
24. Hahn S, Schweins M, Kürner T. Impact of SON function combinations on the KPI behaviour in realistic mobile network scenarios. In: 2018 IEEE wireless communications and networking conference workshops (WCNCW), 2018. P. 1–6.
25. Laselva D, Mattina M, Kolding TE, Hui J, Liu L and Weber A. Advancements of QoE assessment and optimization in mobile networks in the machine era. In: 2018 IEEE wireless communications and networking conference workshops (WCNCW), Barcelona, Spain, 2018. P. 101–6.
26. Verzani J. Getting started with RStudio. California: O'REILLY; 2011.
27. VMware. Virtualization Overview. 2006. <https://www.vmware.com/pdf/virtualization.pdf>. Accessed 05 Sept 2018.
28. Frampton M. Big data made easy, a working guide to the complete hadoop to. New York: APRESS; 2015.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)