

Q-score: Proactive Service Quality Assessment in a Large IPTV System

Han Hee Song[§], Zihui Ge[‡], Ajay Mahimkar[‡], Jia Wang[‡], Jennifer Yates[‡], Yin Zhang[§],
Andrea Basso[‡], Min Chen[‡]

The University of Texas at Austin[§]
Austin, TX, USA

AT&T Labs – Research[‡]
Florham Park, NJ, USA

{hhsong,yzhang}@cs.utexas.edu {gezihui,mahimkar,jiawang,jyates,basso,mc4381}@research.att.com

Abstract — In large-scale IPTV systems, it is essential to maintain high service quality while providing a wider variety of service features than typical traditional TV. Thus service quality assessment systems are of paramount importance as they monitor the user-perceived service quality and alert when issues occurs. For IPTV systems, however, there is no simple metric to represent user-perceived service quality and Quality of Experience (QoE). Moreover, there is only limited user feedback, often in the form of noisy and delayed customer calls. Therefore, we aim to approximate the QoE through a selected set of performance indicators in a proactive (*i.e.*, detect issues before customers reports to call centers) and scalable fashion.

In this paper, we present a service quality assessment framework, Q-score, which accurately learns a small set of performance indicators most relevant to user-perceived service quality, and proactively infers service quality in a single score. We evaluate Q-score using network data collected from a commercial IPTV service provider and show that Q-score is able to predict 60% of the service problems that are reported by customers with 0.1% false positives. Through Q-score, we have (i) gained insight into various types of service problems causing user dissatisfaction, including why users tend to react promptly to sound issues while late to video issues; (ii) identified and quantified the opportunity to proactively detect the service quality degradation of individual customers before severe performance impact occurs; and (iii) observed possibility to allocate customer care workforce to potentially troubling service areas before issues break out.

Categories and Subject Descriptors

C.4 [Computer-Performance of Systems]: Reliability, availability, and serviceability

General Terms

Management, Reliability

Keywords

IPTV, Service, Quality, QoE, Assessment, Analysis, Video

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IMC'11, November 2–4, 2011, Berlin, Germany.

Copyright 2011 ACM 978-1-4503-1013-0/11/11 ...\$10.00.

1. INTRODUCTION

IPTV technologies are transforming the global television industry, enabling new operators to provide TV services whilst also enabling a wealth of innovative new IP-based services integrated with more traditional TV. However, there is a pressing need to ensure that the IPTV services being deployed deliver a Quality of Experience (QoE) that is equal to or better than traditional TV services.

The traditional approach to assessing quality of experience (QoE) has been through subjective evaluation in controlled laboratory environments. Unfortunately, subjective evaluation is expensive, error-prone and unreliable. A complementary approach is through user feedback. It is, in general, collected by the call centers and it provides a direct measure of the service performance problems experienced by the users. However, user feedback is not always complete (not all users report service quality issues) and delayed (users have already been negatively affected by the time they call customer care center to report their service quality issues).

What operators lack today is a proactive approach to obtaining comprehensive views of user's quality of experience. Such views are critical to proactively detecting service issues that matter to customers, so that operators can rapidly respond to them to ensure a high-quality customer experience. Without such visibility, operators risk being unaware of service degradations - instead they learn about issues only when customers reach frustration points and report to customer care call centers. Thus, proactive assessment of quality of experience is crucial to providing operators with insights into the ultimate metric - what customers are experiencing - so that they can effectively manage their service offerings and detect and ideally respond to issues *before customers report issues*. Proactive assessment of quality of experience can also help operators to effectively dimension the customer care workforce in anticipation of the large volume of user calls and customer-impacting conditions can be avoided.

Although there is extensive monitoring of network elements in place today and operators rapidly react to issues which are reported by the network elements - there is no technology which can directly measure *customer perception* of TV service quality. Video monitoring technologies exist, but it is still non-trivial to relate such measurements to customer perception. Deploying video monitoring to millions of customers is also prohibitively expensive, and service providers are typically constrained by the technology available within the Set-Top Boxes.

In the absence of direct measurements, we instead focus on using network measurements to infer customer service experience. However, such an approach is still challenging, due to (i) incomplete knowledge about the relationship between user-perceived issues and network performance metrics, and (ii) user feedback about

quality of experience is biased towards the negative (i.e., customer calls on reporting issues) and is often delayed, noisy and limited.

In this paper, we propose a new framework, which we refer to as *Q-score*, for proactive assessment of user perceived quality of experience. *Q-score* constructs a single quality of experience score using performance metrics collected from the network. *Q-score* consists of two key components: (i) offline learning of the association between the service quality of experience and the network performance metrics collected from the servers, routers and in-home equipment, and (ii) online computation of the score for individual users or groups of users. *Q-score* captures the quality of experience by users in a timely fashion and can provide operators with rapid notification of service issues, often giving them a lead time of several hours before the user reports to the call center.

Q-score Design Challenges. Due to the interwoven server and network system, as well as the sophisticated hardware and software composition of home network devices, assessing service quality of experience is a sophisticated task. The proposed *Q-score* approach uses customer reports (e.g., tickets) to provide feedback regarding issues that customers are concerned about. Designing *Q-score* required us to address the following key challenges:

1. **Associating QoE with network performance.** Because of the inherent difference between network-level performance indicators and user-perceived quality of service, associating the two does not occur naturally. Even for domain experts, since there is no objective video quality metric, it is not trivial to identify key performance indicators that are closely related to quality of experience. Even if the metric were available, it would require more finely grained monitoring of network indicators, which in turn might introduce scalability issues.
2. **Lack of timely, high-quality user feedback.** User feedback is inherently noisy, incomplete and delayed. Depending on situations such as the individual viewer's living schedule, the severity of the issue, there are large variances between the beginning of service quality issues and reporting times. Some users issue a report immediately after they observe a service quality degradation; others may wait hours before calling customer service centers. Similarly, different users have different tolerance levels to service quality issues - one user may report incessantly regarding issues that another user may barely notice. This all makes it inherently challenging to use such feedback to associate service quality of experience with network performance.
3. **Large volume of diverse performance measurements.** From a network perspective, service providers typically collect fine-grained measurements from the routers and servers (e.g., real-time syslogs, and regular polls of SNMP performance counters such as CPU, memory, packet counts, and losses). Some performance measurements inside the home may be fine-grained (e.g., residential gateway events), whereas others may be coarse-grained (e.g., hourly or daily summaries of Set-Top Box events). Set-Top Box (STB) data collection is intentionally not fine-grained to minimize the potential of service disruption due to measurements and due to the massive scale of the measurement infrastructure that would be required. The diversity in the granularity of performance measurements poses interesting technical challenges in inferring the quality of experience.

Our Contributions.

1. We design and implement a prototype *Q-score* system for proactively assessing quality of experience for IPTV users. *Q-score* uses a multi-scale spatio-temporal statistical mining technique for computing a single score capturing the quality of experience. By performing spatio-temporal aggregation and multi-scale association of the user feedback with network performance metrics, it identifies the right set of metrics useful for accurately quantifying the quality of experience.
2. We evaluate *Q-score* using data collected from a large commercial IPTV service provider and show that *Q-score* is able to predict 60% of customer service calls with 0.1% of false positives.
3. We create three applications above *Q-score*. (i) *Identifying important Key Performance Indicators (KPIs)* that are statistically associated with the quality of experience, (ii) *Predicting bad quality of experience* to users and generating alerts to the Operations team, and (iii) *Effective dimensioning of the customer care workforce* to dynamically allocate repair personnel to service regions as they experience issues for conducting root-cause diagnosis and rapid repair.

Organization. The remainder of the paper is organized as follows. In Section 2, we provide background information regarding the IPTV network architecture and its data. We describe the design of *Q-score* in Section 3, with details on its offline learning component and online monitoring component. We present performance evaluation results in Section 4. In Section 5, we explore three important applications of *Q-score*. We review related work in Section 6 and offer conclusions in Section 7.

2. BACKGROUND

In this section, we give an overview of the IPTV service architecture followed by a detailed description of the data sets used in the paper.

2.1 IPTV Service Architecture

Figure 1 provides a schematic overview of an IPTV system. The service network exhibits a hierarchical structure where video content is delivered via IP multicast from servers in the provider's core network to millions of Set-Top Boxes (STBs) within home networks. Specifically, either the Super Head-end Office (SHO) which serves as the primary source of national content or Video Head-end Offices (VHOs) which governs local content at each metropolitan area encodes, packetizes and sends the content towards end users. Depending on the service provider, the video content goes through several routers and switches in Intermediate Offices (IOs), Central Offices (COs), a Digital Subscriber Line Access Multiplexer (DSLAM), and a Residential Gateway (RG) before reaching STB where the packetized content gets decoded and displayed on the TV. All of the network entities comprising the IPTV service logs Key Performance Indicators (KPIs) such as delivery status of data and health diagnostics.

2.2 Data Sets

In the paper, we use data collected from a large commercial IPTV service provider in the United States, which has customers spread throughout four different time-zones. Our data set consists of (i) network performance indicators, (ii) user behaviors and activities, and (iii) user feedback in the form of customer trouble tickets

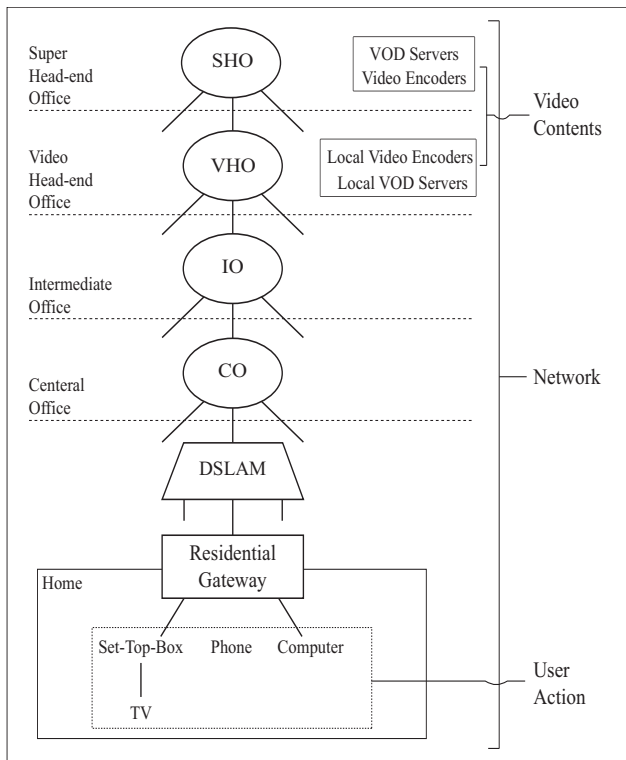


Figure 1: IPTV service architecture

as summarized in Table 1. We normalize timestamps in all data sets to GMT to accurately and effectively associate the user feedback with performance metrics and user behaviors. The data has been collected for 45 days from August 1st to September 15th, 2010. Note that all the information related to the users are anonymized to preserve their privacy.

Network Performance Indicators. Network performance indicators are categorized into two types: (i) provider network performance indicators, which are collected from routers and switches in SHO, VHO, IO, CO of the IPTV service provider as shown in Figure 1 and (ii) home network performance indicators, which are collected from components inside user’s homes (i.e., RG and STB). For the provider network performance data, we collected SNMP MIBs from every router and switch in SHO, VHO, IO, and CO. The SNMP MIBs report five minute average performance statistics of CPU utilization and fifteen minute average summaries for packet count, packet delivery errors and discards.

From the home network side, we first collected data relevant to each STB and RG. Each STB records audio and video streaming-related information including video throughput, receiver transport stream errors, codec errors, DRM errors, and viewing duration of TV channels. The video streaming-related information is reset when the TV tuner clears its buffer by switching channels. While each STB logs all the TV viewing information at all times, polling servers only take a subset of the STBs’ statistics at each polling interval (due to the high volume of audio and video log and traffic overhead during data delivery). As a result, only a sampled set of STBs (i.e., 2% of all STBs) are used in our study. Secondly, we collected STB syslog information that contains a wide variety of hardware and software information, such as hard disk usage and memory usage, data delivery status such as packet error rate and buffer usage. The diagnostic information are collected in the same way as the

Data Set Type	Spatial Level	Description
Network Performance Indicators	STB	STB audio quality indicators
		STB video quality indicators
		STB syslog
		STB resets
	STB crashes	
	RG	RG Reboots
User Activity Indicators	User	IO & CO
		SHO & VHO
User Feedback	User	STB power on/off log
		STB Channel change log
		STB Stream control log
Customer trouble tickets		

Table 1: Summary of data sets

STB audio and video log, i.e., sampled data were polled by collection server in round robin fashion. Thirdly, we collected crash and reset events log from each STB. The crash events log refers to unexpected rebooting of STBs due to software malfunctions and the reset refers to intentional and scheduled reboots commanded by network operators due to, for instance, software updates. The crash and reset log are periodically collected from all STBs with millisecond scale time stamps. Last performance indicator taken from home network is the reboot log of RGs that are commanded by operators remotely. RG reboot logs are collected in the same way as STB reboot logs. The crash and reboot logs are collected from the entire seven million STBs.

User Activity Indicators. Because IPTV networks are highly user-interactive systems, certain user activity patterns or habits can create overload conditions on the STB and cause a service issue (e.g., a user changing channels too frequently may cause its upstream device such as a DSLAM to be overwhelmed, leading to an inability to handle all of the remaining STBs that it serves). Hence, user activities are another important factor to be considered. Similar to conventional TV users, IPTV users use a vendor/provider-customized remote controller to control the STB. For this paper, we collected logs from every STB that captures four types of user activities performed: (i) power on/off: this is the result of the user pressing the power button to turn on or off the STB; (ii) channel switch: this can be the result of one of the three actions: target switching by directly inputting the channel number, sequential scanning by pressing the Up/Down button, or pre-configured favorite channel list; (iii) video stream control: this includes actions such as fast forward, rewind, pause and play that are performed on either live TV streams, VoD, or DVR; and (iv) on-screen menu invocation: this log saves the user action of pulling up the STB menu displayed on TV screen that lets the users to access the features provided by the IPTV system.

User Feedback. For user feedback, we used calls made to the customer care center of an IPTV service. Customer care cases are records of user interactions at call centers. A customer call can be related to service provisioning, billing and accounting, or service disruption. Since the focus of our paper is on quality of experience (QoE), we specifically examined users’ reports on service disruptions that later involved technical support as our user feedback. Each customer call related to service disruption includes the anonymized user ID, report date and time, brief description of the problem, and resolution of the issue.

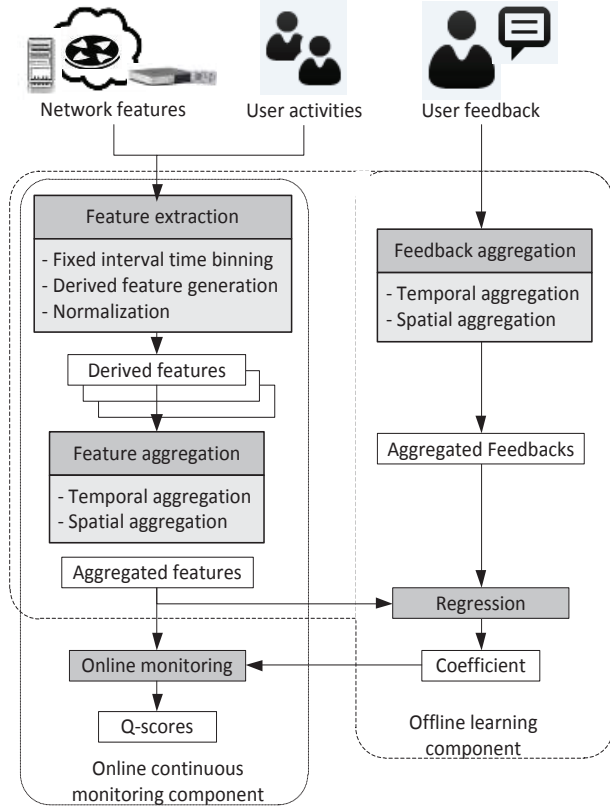


Figure 2: Overview of Q-score design

3. Q-SCORE DESIGN

In this section, we introduce our proposed scheme, Q-score. The high level idea is to extract a useful association between the noisy, incomplete, and indeterminately-delayed user feedback and the various network (including the servers, transport and in-home devices) performance indicators through an offline learning process, and then transform the knowledge into an online monitoring system that estimates/predicts user-perceived service quality based on the available network KPIs. We start by giving an overview of the Q-score system architecture and then dive into details of each component.

3.1 Overview

Figure 2 presents the system architecture of Q-score. As shown in the figure, Q-score takes input from network (including servers, transport and in-home devices) performance indicators, which we refer to as *features*, the user control activities, and the user feedback in the form of customer call service records. The output is a series of Q-scores, one for each user of the service, quantifying the received service quality. At a high level, our system is composed of two components: (i) an offline learning component and (ii) an online continuous monitoring component. The overall dataflow in Q-score system begins with the offline relationship learning between user feedback on service quality and the measurements from the network features and user activities. Ideally, if there had been any accurate and fine-grained user-level service quality measure, we would use it to train a model for network feature selection. However, as stated earlier, the best available method for discovering user-level service quality issue is through the lossy,

noisy and indeterminately-delayed calls to customer care centers. Consequently, we need to carefully design the appropriate temporal and spatial aggregations to remedy the inherent loss, noise and delay with user feedback. Furthermore, by applying statistical regression over a large quantity of historical data between various network KPIs and the user feedback, we obtain a set of regression coefficients which quantitatively capture their relationship. These regression coefficients are fed into the online monitoring component.

With the regression coefficients, we can turn the up-to-date network KPI measurements into a single numerical score for each user or groups of them within a given spatial region. The numerical score, which we refer to as the *Q-score*, captures the likelihood of any on-going service quality problem. Tracking the Q-score over time enables many service management applications, as will be described in Section 5.

3.2 Spatio-Temporal Feature Extraction

In order to discover possible correlation between user’s quality of experience and IPTV system events, we apply a comprehensive set of performance indicators ranging from provider network performance indicators to home network component status logs, and to user interaction logs with IPTV. On each of the network performance indicators and user interaction indicators described in Section 2.2, we apply the following series of transformations to obtain a measurement matrix.

3.2.1 Transformations of Measurement Readings

Conversion to Fixed-Interval Time Bins. Network measurement data collected from different sources and devices are bound to different time periods, posing challenge in correlating them. Some data sets, such as CPU level of routers in SNMP MIBs, contain periodically collected measurement data, and the value represents the average or total over the measurement interval. Some other data sets, such as user activities to STB and STB crash logs, contain events that take place at a single point of time, hence are intermittent and have zero duration. Data sets such as STB audio and video quality indicators contain data polled either on demand or at irregular intervals and represent the cumulative counters over a variable time interval (e.g., due to channel switches clearing the diagnostic counter entries).

To unify the data representation, we define a data point $d(m, l, s, e) = v$ as composed in a four dimensional specification: (i) metric $m \in M$ where M is a set of metrics such as CPU level of routers and count of video decoding errors at STBs. (ii) location $l \in L$ where L is a set of spatial location identifiers such as a set of users, DSLAMs, or COs. (iii) beginning time for the data binding interval $s \in T$, where T is the total time window, and (iv) ending time of the data binding interval $e \in T$. v is the measurement value that d contains. Note that for measurement data pertaining to a single time point, $s = e$.

The above representation is comprehensive in capturing various cases of periodic/intermittent or fixed/variable duration measurements. However, it requires a moderate amount of computation to determine the overlaps among the time intervals, which becomes prohibitively expensive for a large data set as in our case. To reduce the complexity, we convert all $d(m, l, s, e)$ into a fixed-size time interval data representation $b(m, l, s, \delta)$ as follows:

$$b(m, l, s, \delta) = \{v \mid v = d(m, \bar{l}, \bar{s}, \bar{e}), \text{ where } l = \bar{l} \text{ and } [\bar{s}, \bar{e}] \text{ overlaps with } [s, s + \delta]\} \quad (1)$$

where δ is length of the feature time interval. Note that if there exist two or more ds with matching measurement time to $[s, s + \delta]$, there could also be multiple identical values for b – making b not well defined. We need to introduce the aggregation functions as below.

Conversion to Derived Features. To deal with multiple ds colliding into the same b (either due to time bin or spatial aggregation), we define three types of aggregate data points, which we refer to as the *derived features*. They contain (i) the minimum, (ii) the maximum, and (iii) the average of all the values for b respectively. Formally,

$$f_m(m, l, s, \delta) = \min_{l \in \text{child}(\bar{l})} (\cup(b(m, \bar{l}, s, \delta))).$$

$$f_M(m, l, s, \delta) = \max_{l \in \text{child}(\bar{l})} (\cup(b(m, \bar{l}, s, \delta))). \quad (2)$$

$$f_a(m, l, s, \delta) = \text{avg}_{l \in \text{child}(\bar{l})} (\cup(b(m, \bar{l}, s, \delta))). \quad (3)$$

In this way we can limit the number of derived features to be three regardless of the number of actual readings in b . Unless specified otherwise, all features referred in the rest of the paper are the derived features.

Feature Normalization. To identify a small set of network features most relevant to customer feedback, we need to fairly compare each network feature to others. However, the network features we consider typically take numerical values, potentially having different signs and across large range of scales. This makes it difficult to assess the significance of their associated coefficient under regression.

To deal with the diverse data values, we further normalize features to be binary-valued by comparing to a threshold, which is determined depending on the metric and location.

Consider a vector of features of the same metric and location over different time and interval combinations:

$$\vec{f}_a(m, l) = \langle f_a(\bar{m}, \bar{l}, s, \delta) \text{ where } m = \bar{m}, l = \bar{l} \rangle \quad (4)$$

We need to identify a threshold value τ for \vec{f}_a . To do so, we bring in the user feedback in the form of user call logs. We consider the conditional distribution function of the metric value of interest when (1) there is one or more entries of the user call log being associated with the location l and when (2) there is no such entry. Ideally, a threshold τ can separate the instances between cases 1 and 2. When threshold τ is low, the chance of having instances in case 1 passing the threshold increases, and when the threshold is high, the chance of having instances in case 2 failing the threshold increases. So, we set the threshold τ such that the two factors balance out. Using empirical CDFs of the case 1 (F_1) and case 2 (F_2), we can define τ to be the intersecting point of F_1 and $1 - F_2$ such that

$$F_1(\tau) = 1 - F_2(\tau). \quad (5)$$

Once τ is determined, we can normalize of f_a as follows.

$$f_a(m, l, s, \delta) = \begin{cases} 1 & \text{if } f_a(m, l, s, \delta) \geq \tau \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Features f_m and f_M can be normalized in the same way.

3.2.2 Constructing Measurement Matrix

In order to support multi-scale analysis that accounts for the indeterminate delay in user feedback, we construct the regression input matrix \mathbf{X} over all measurement metrics, location, and time parameters as below.

$$\mathbf{X} = \begin{bmatrix} \begin{bmatrix} f_m(m_1, l_1, s_1, \delta) f_M f_a \\ f_m(m_1, l_1, s_2, \delta) f_M f_a \\ \vdots \\ f_m(m_1, l_1, s_t, \delta) f_M f_a \\ f_m(m_1, l_2, s_1, \delta) f_M f_a \\ f_m(m_1, l_2, s_2, \delta) f_M f_a \\ \vdots \\ f_m(m_1, l_2, s_t, \delta) f_M f_a \\ \vdots \end{bmatrix} & \begin{bmatrix} f_m(m_2, l_1, s_1, \delta) f_M f_a \\ f_m(m_2, l_1, s_2, \delta) f_M f_a \\ \vdots \\ f_m(m_2, l_1, s_t, \delta) f_M f_a \\ f_m(m_2, l_2, s_1, \delta) f_M f_a \\ f_m(m_2, l_2, s_2, \delta) f_M f_a \\ \vdots \\ f_m(m_2, l_2, s_t, \delta) f_M f_a \\ \vdots \end{bmatrix} & \dots \end{bmatrix} \quad (7)$$

The columns of \mathbf{X} represent different metrics of derived features. Thus, each column has f with a unique m_i , where i is an instance of time bins. The rows of \mathbf{X} represent all feature values during a specific time (s_i, δ) at a specific location l_j . Assuming there are n locations, t different time bins, and k different KPI metrics and feature aggregations, the number of rows in \mathbf{X} is $n \times t$ and the number of columns is k .

3.2.3 Multi-scale Temporal Level Aggregations

The time window parameter δ plays an important role in capturing the extend of cause-effect delays. Large δ would include cause-effect relationship with long delay. However, large δ would make it insensitive to dense measurements with short cause-effect delay, as the aggregation weakens the significance of correlation. Since different δ values have advantages over others, we adopt a multi-scale analysis approach by including multiple time window parameters into our consideration. Our matrix representation in Eq (7) is flexible enough to enable this – we append in columns the $\mathbf{X}(\delta_i)$ s with different time-intervals (δ_i).

$$\mathbf{X}_{\text{Temp.Comb.}} = [\mathbf{X}(\delta_1) \cdots \mathbf{X}(\delta_v)] \quad (8)$$

where v is the number of different values of the time window parameter.

3.2.4 Multi-scale Spatial Level Aggregation

Similarly to the temporal aggregation captured by the time window parameter, there can be multiple spatial aggregation levels with an IPTV system architecture. Based on the hierarchical structure in Figure 1, we consider three different spatial aggregation levels in Q-score, namely user, DSLAM, and CO levels.

Single-Scale Spatial Level Aggregation. We set the baseline spatial aggregation level to per-user aggregation. This is because the customer service report logs are associated with a household, which we loosely refer to as a user. Matching the network features to the household/user level, one of the following process is necessary: (i) for features at finer grained spatial level than user (such as STB related features since one household may have multiple STBs), we take the maximum among different feature values for the more specific locations as the representation for f_M , the minimum for f_m , and the average for f_a , at the user level; (ii) for features with coarser grained spatial level than user (such as DSLAM and CO), we replicate the coarser grained feature values for each associated user within the hierarchy. In this way, we preserve the number of samples to be $n \times t$ in each row of \mathbf{X}_{user} . The same spatial level aggregation is applied for the DSLAM level and the CO level to obtain $\mathbf{X}_{\text{DSLAM}}$ and \mathbf{X}_{CO} respectively.

Multi-Scale Spatial Level Aggregation. In parallel with the multi-scale analysis with respect to time window parameter, different spatial aggregation levels can be fed into regression altogether. The idea is that the most prominent feature would be at a suitable spatial aggregation level and would dominate the same features aggregated at other spatial levels. We append in column the feature matrices for different spatial levels to get the $\mathbf{X}_{\text{Spat. Comb.}}$:

$$\mathbf{X}_{\text{Spat. Comb.}} = [\mathbf{X}_{\text{userID}} \ \mathbf{X}_{\text{DSLAM}} \ \mathbf{X}_{\text{CO}}]. \quad (9)$$

3.3 Feedback Aggregation

As outlined in Section 2.2, we use the customer service call logs as the user feedback regarding service quality. This feedback is inherently unreliable. It is incomplete as not all service quality problems (e.g., video glitches) would be noticed and reported by users. And there is an indeterminate delay ranging from minutes to hours to even days between the service problem and the trouble ticket log entry (*i.e.*, entries of customer reporting issues to call centers). All of these require some denoise processing for such user feedback to be useful even in statistical sense.

We adopt the same principle applied in the spatio-temporal aggregation with respect to network features. Let c be the predicate of the presence of a matching entry in the feedback log (B):

$$c(l, u, \gamma) = \begin{cases} 1 & \text{if } \exists b \in B \text{ during } [u, u + \gamma]; \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

where u is the beginning time for a feedback binding interval and γ is the length of feedback time interval. Once $c(l, u, \gamma)$ is defined, we can use the same spatio-temporal aggregation method for the network features on c .

A network event or user activity is always a cause of user feedback but cannot be an effect. Thus we set $u = s + \delta$ so that when we correlate c_i to b_i , we take account of the causal sequence between network (or user activity) events and user feedback. Let y be a vector of feedback for different users over time

$$\mathbf{y} = [c(l_1, u_1, \gamma), \dots, c(l_1, u_t, \gamma), c(l_2, u_1, \gamma), \dots, c(l_2, u_t, \gamma), \dots]^T.$$

The length of the vector \mathbf{y} is determined by the number of locations n and the number of time bins t , making it to be $n \times t$ which is the same as the row count of \mathbf{X} .

3.4 Regression

Given the measurements of network indicators \mathbf{X} and user feedback \mathbf{y} , we now aim to find a coefficient vector β that provides a compressed representation of the relationship between \mathbf{X} and \mathbf{y} . Note that, in the event of measurement or data collection error which results in parts of \mathbf{X} or \mathbf{y} to have no values, we remove the affected rows of \mathbf{X} and \mathbf{y} from consideration to eliminate possible false correlation.

Such an optimization can be performed using *regression*. A baseline regression model of linear regression [9], however, cannot provide the optimal solution as our system of equation $\mathbf{X}\beta = \mathbf{y}$ is over-constrained (*i.e.*, the equation has far smaller number of unknowns than the number of equations ($k \ll (m * n)$)). To prevent β from over-fitting due to high variance, we apply Ridge regression [11] that imposes a penalty λ on the complexity of model by minimizing a penalized residual sum of squares \mathcal{RSS} as follows

$$\min_{\beta} \mathcal{RSS}(\mathcal{D}, \beta) \text{ s.t. } \sum_{i=1}^n \beta^2 \leq s. \quad (11)$$

where \mathcal{D} is the set of observed data points $\mathcal{D} = x_n, y_n$.

We can state this optimization problem in Ridge regression as

$$\hat{\beta} = \arg \min_{\beta} \sum_i (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2. \quad (12)$$

The Ridge coefficient $\hat{\beta}$ becomes

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (13)$$

where \mathbf{I} is the identity matrix.

There are other regression methods we have explored including l_1 -norm minimization and logistic regression. However, as our system of equation has tens of thousands of equations and thousands of unknowns, l_1 -norm minimization and logistic regression either took excessive amounts of time in computation or failed to converge to an answer. The complexity and scale of our system make these other techniques infeasible.

Finding Significant KPI Weights. From the β coefficients, we can identify key performance indicators (KPIs) that are more closely related to user feedback. This involves sorting the regression coefficients by their absolute value and identifying the top N KPIs associated with them. Furthermore, by analyzing the commonality and difference of the same metric across different temporal and spatial aggregation configuration, we can gain insight on how each of these KPIs impact the users' quality of experience specific to the most significant spatio-temporal aggregation. The analytical results on the most significant KPIs in IPTV are presented in Section 5.1.

3.5 Compute Q-score in Runtime

Once the offline learning of β completes, we can compute from the available key performance indicators the Q-scores either for individual users or groups of users aggregated spatially depending on the feedback aggregation scheme used.

Detecting Significant Q-score Changes. We apply β from the offline learning to the current network measurement data \mathbf{X} and obtain Q-score that estimates per-user level service quality. Running continuously as network KPI data streaming into Q-score, we track the series of Q-scores over time. Since Q-scores are real-valued numbers, we need to identify the thresholds for alarming on the Q-scores to the operations. The alarms can be proactively used to predict customer calls. We apply simple threshold-based change detection on the time-series of Q-scores to generate the alarms.

False alarm rate of Q-score. As a prediction mechanism of possible outbreaks, it is very important to have a low false alarm rate in a service quality assessment. In Q-score, a multitude of components prevent a single user, one end-user device, or a network device from raising false alarms for a large population of users. The feature normalization described in Section 3.2.1 regulates feature values, an exceptional feature value for an individual cannot affect much to the entire population. The multi-scale aggregations (Section 3.2.3, 3.2.4) further reduces the possibility of falsely emphasizing rare events. In the case of spatial aggregation, because Q-score considers both individual users and spatial groups of users, the score is stable even when an individual's feature value is high. Similarly, temporal aggregation prevents the chance of false alarms due to highly transient feature value changes. Additionally, in practice, we carefully set the threshold of Q-scores to focus on minimizing false positives, even with slight sacrifice to coverage (recall).

4. EVALUATION

In this section, we present the performance evaluation results of Q-score and show that the regression results are accurate and robust, and the multi-scale aggregation of spatio-temporal features has benefit over single scale, non aggregated cases.

4.1 Evaluation Methodology

Metrics. We compare the number of predicted customer trouble tickets and that of received customer trouble tickets and measure the accuracy of prediction of service quality issues by false negative rate (FNR) and false positive rate (FPR). The FNR and FPR are computed per user basis.

$$FNR = \frac{\text{\#of time bins that Q-score fails to predicts a trouble ticket}}{\text{\#of time bins that have received trouble tickets}}$$

$$FPR = \frac{\text{\#of time bins that Q-score incorrectly predicts a ticket}}{\text{\#of time bins that do not have any trouble tickets}}$$

Note that due to the sparsity in the occurrence of user feedback (i.e., trouble tickets), the number of time bins without any user feedback is orders of magnitude higher than the number of time bins with user feedback.

Training and Testing Sets. In our evaluation of the Q-score system, we use data sets collected from a commercial IPTV network provider in US over two months time period from August 1st, 2010 to September 30th, 2010. Unless otherwise mentioned, we use 15 days of data collected from August 15th, 2010 to August 29th, 2010 as the training data set for β and the subsequent 15 days of data collected from September 1st, 2010 to September 15th, 2010 as the testing data set. In addition, we use multi-scale temporal aggregation of $X_{Temp.Comb.}$ combining δ of 3-24 hours and multi-scale spatial aggregation of $X_{Spat.Comb.}$ combining spatial levels of user, DSLAM, CO, and VHO as the default setting. Lastly, we set the default feedback time bin γ to be $\gamma = 24$ hours.

We assign λ a small positive value within $(0, 0.05]$. While different λ exhibit small differences in accuracy, the optimal λ varied from data set to data set. Since the selection of λ is specific to data set in each test, we present the results with the best λ while omitting to show its actual value.

4.2 Results

4.2.1 Accuracy Analysis

We begin our evaluation by assessing how well Q-score follows the ground truth of user-perceived service quality. In our evaluation, we use user feedback as an approximation of the ground truth of user-perceived service quality issues in training and testing Q-score system. Recall that the user feedback is incomplete in reflecting user perceived service quality. In fact, the user feedback captures a subset of user perceived service quality issues and thus underestimates the actual occurrences of service performance degradations. Fortunately, major and/or long lasting service performance degradations are likely to be captured by the user feedback [24]. Hence, it is likely that the computed Q-score underestimates the actual user perceived performance issues, but expected to capture major outages and performance degradations.

While Q-score does not perfectly match with the user perceived service quality at the individual user level, the changes or trends in the distribution of Q-score are expected to follow closely with that of the actual service quality degradation at certain spatial aggregation levels. In our evaluation, we choose CO as the aggregation

Aggregation method	P value in F-test	Correlation coefficient \mathcal{R}
CO	0.00	0.6826
Random	2.21e-31	0.7165

Table 2: Accuracy analysis results of Q-score

level¹. By summing up individual users' feedback within each CO into a single value, we obtain an aggregation vector \mathbf{S}_{actual} of user feedback. Since \mathbf{S}_{actual} is a spatio-temporal aggregation of user feedback, its element now signifies the level of user-perceived service quality issues. Similarly, by summing up the individual users' Q-score inside each CO into a single value, we can obtain an aggregation vector of Q-scores \mathbf{S}_{estim} that signifies our estimated level of user-perceived service quality.

To evaluate the significance of the relation between the actual (\mathbf{S}_{actual}) and estimated (\mathbf{S}_{estim}) user perceived service quality level, we run an F-test between them. Let the null hypothesis $H_0 : r = 0$ where $\mathbf{S}_{actual} = r * \mathbf{S}_{estim}$. We find that for the significance level of 0.1, the hypothesis test is rejected, implying that the relation between the two vectors does exist. A Pearson's correlation test also shows relatively high correlation coefficient \mathcal{R} between \mathbf{S}_{actual} and \mathbf{S}_{estim} , proving that the relationship between the two is linear. In other words, Q-score does follow the user-perceived service quality.

Because CO level aggregation represents spatial proximity of user geographical locations, user feedback rates can be different across COs. To evaluate if CO aggregation introduce any bias on the results, we also conduct the same evaluation using a random grouping with the same number of groups as the number of COs and compute aggregation vectors. Table 2 summarizes the F-test and Pearson's correlation tests results for both CO level aggregation and random grouping based aggregation. The random grouping based aggregation generally shows the same results as the CO level aggregation, supporting that Q-score indeed follows user feedback regardless of how we aggregate users in Q-score computation.

4.2.2 Multi-scale Temporal Aggregation

In this section, we evaluate the impact of different time-bin size (δ) on network indicators (single-scale temporal level aggregation). Then we show the performance benefits by using multi-scale temporal aggregation on network performance indicators (multi-scale temporal level aggregation).

Figure 3 shows the Q-score on FPR-FNR trade-off curves using various δ s ranging from 3 hours to 24 hours (i.e., each curve corresponds to an \mathbf{X} with a given δ). Note that FPR shown on the x -axis is in log-scale and FNR shown on the y -axis is in normal scale. The figure shows that the prediction accuracy gets generally better as we shorten δ (i.e., the curve gets closer to the lower left corner of the plot). However, comparing $\delta = 3$ hours and $\delta = 6$ hours, their FNR overlaps over different range of FPR, indicating that there is no single optimal δ to be chosen.

Figure 4 shows the results of $\mathbf{X}_{Temp.Comb.}$ by applying multi-scale temporal aggregation on network performance indicators. There are three curves obtained by combining (i) shorter time bins of 3-12 hours, (ii) longer time bins of 15-24 hours, and (iii) the entire range of 3-24 hours. We observe that (iii) provides the best performance among them. At the same time, (iii) is also strictly

¹We considered various levels of spatial granularity in the IPTV hierarchy including DSLAM, CO, and VHO levels. Among them, CO level aggregation is most adequate for the accuracy analysis because it yields a statistically sound number of user IDs in each CO and enough number of COs to make meaningful comparisons between aggregation vector Ses .

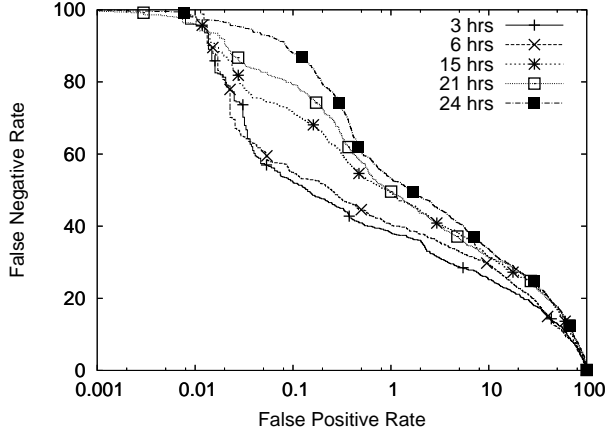


Figure 3: Comparison of various δ_s on features (performance indicators)

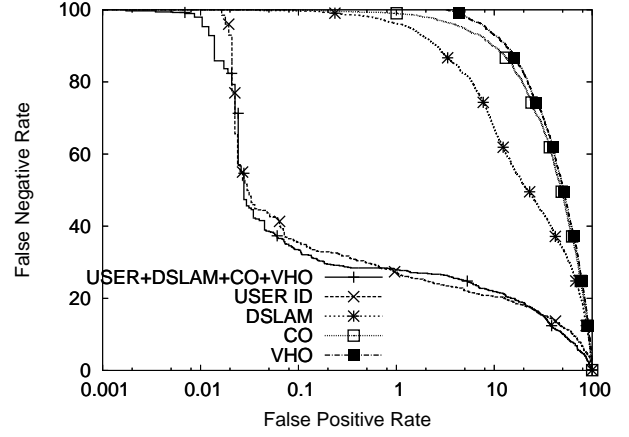


Figure 5: Comparison of various spatial aggregation levels on features (performance indicators)

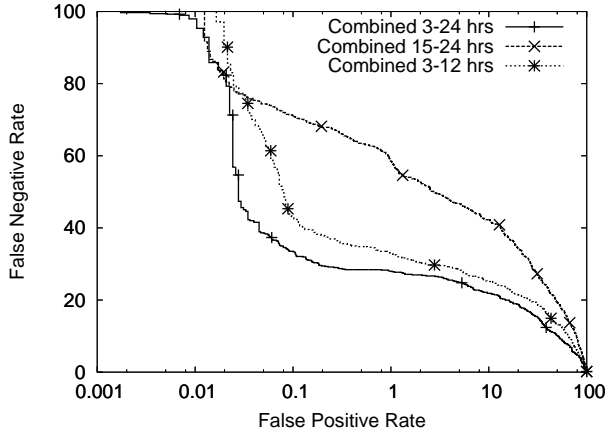


Figure 4: Comparison of multi-scale temporal aggregations on features (performance indicators)

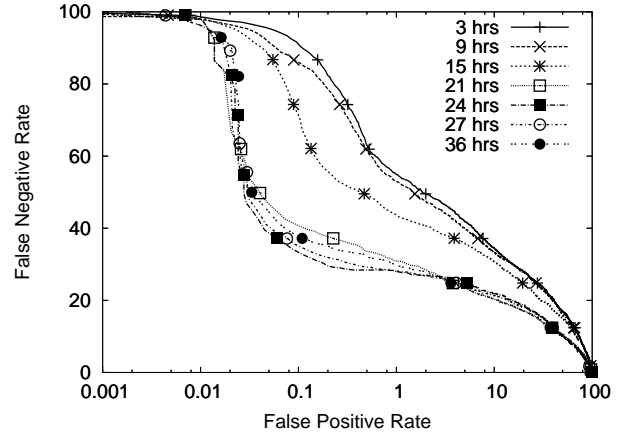


Figure 6: Comparison of various time bin size γ on user feedback

better than any curves in Figure 3, proving that multi-scale temporal aggregation on network performance indicators does improve the accuracy of Q-score prediction on service quality issues.

4.2.3 Multi-scale Spatial Aggregation

We now evaluate the impact of various levels of spacial aggregation on network performance indicators and the benefit of using multi-scale spatial aggregation in Q-score.

Figure 5 shows the trade-off curves of \mathbf{X} with various single-scale spatial aggregation ranging from user ID ($\mathbf{X}_{\text{userID}}$), to DSLAM ($\mathbf{X}_{\text{DSLAM}}$), to CO (\mathbf{X}_{CO}), and to VHO (\mathbf{X}_{VHO}) level. As the spatial aggregation level changes from user ID to DSLAM (*i.e.*, smaller-sized region to larger-sized region), we observe that the FNR increases from 35% to 100% when FPR is at 0.1%. A possible explanation to this is that if the service quality issues reported by users are more related to a home network problem rather than a provider network problem, spatial aggregation of network performance indicators can attenuate signals relevant to the individual users at home network side. As we will show in Section 5.1, by analyzing significant KPIs, we are able to confirm that the significant KPIs are mostly related to STB and RG (*i.e.*, home network devices) while backbone network appeared to be well pro-

visioned. In addition to the single-scale spatial aggregation, the first plot of Figure 5 (denoted as ‘USER + DSLAM + CO + VHO’) shows multi-scale spatial aggregation (with measurement matrix $\mathbf{X}_{\text{Spat.Comb.}}$). We observe that the multi-scale spatial aggregation outperforms any single-scale aggregation in terms of overall prediction accuracy, proving that the regression algorithm makes the most accurate selection of spatial level of features.

4.2.4 Feedback Aggregation

To show the effect of user feedback duration being aggregated together, Figure 6 compares various lengths of γ . We observe that as γ gets longer, the regression performance gets better. An explanation for this is, as mentioned in Section 3.3, there is a significant delay between the occurrence of a problem and the filing of user feedback. Due to the elongated delay, time-bins with short γ s may fail to contain feedback correlated with significant network indicator values.

4.2.5 Sensitivity to Training Duration

Finally, we evaluate the sensitivity of testing accuracy on the duration of training. In this experiment, we fix the testing duration and assess how accuracy changes by varying the training duration.

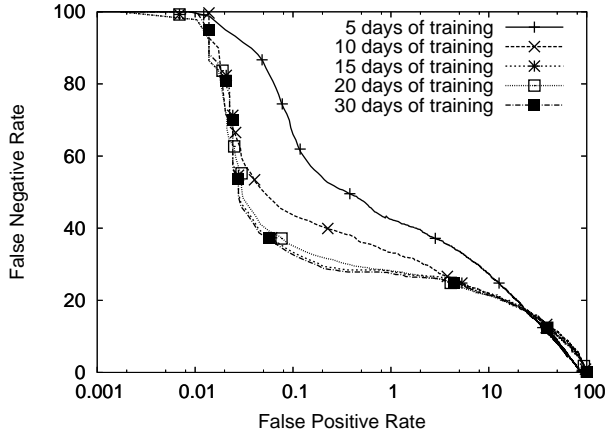


Figure 7: Comparison of accuracy with various training durations

Table 3 shows the dates of training and testing periods used in our evaluation. Figure 7 shows the accuracy trade-off curves of using different training durations. We observe that in general, the testing accuracy improves as we increase the training duration. However, the gain becomes marginal once the training duration is longer than 15 days. This result suggests that using 15 days as training period is a good choice.

A closer examination of the curves corresponding to the use of 15 and 20 days of training duration reveals that the accuracy of using 15 days training duration is marginally better. A possible reason for this is that in the month of August, there was a network-wide STB firmware upgrade. The upgrade that took place between 08/10/2010 and 08/14/2010 could have obstructed measurement of STB logs (*i.e.*, STB audio and video quality measurement logs, syslog, reset and crash logs) and caused learning of β to be affected. Since this kind of glitches occurs in real data, we take small amount of noise as granted. In all, we observe that 15 days of training is enough to learn β .

Summary. In this section, we evaluate the accuracy and robustness of Q-score. The Q-score, combined with multi-scale temporal aggregation and multi-scale spatial aggregation, successfully predicts 60% of service problems reported by customers with only 0.1% misclassification (*i.e.*, false positive rate). While an in-depth analysis is in order, our preliminary test shows that a portion of the remaining 40% of unpredicted issues are either (i) not captured by any of the network KPIs we measure (*e.g.*, remote controller malfunction, wiring issues between STB and TV inside home) or (ii) fallacies that our regression does not capture (*e.g.*, gradual and long term changes in network KPIs). For (i), we are unable to completely ignore such issues as feedback is logged by human operators in plain text. For (ii), we address with our previous works Giza [18] and Mercury [19] as they are specifically designed to detect and mitigate recurring and persistent events in application service networks. In a future work, we plan to conduct an extensive analysis on the false negatives to determine the proportions of the issues in each of the categories and further improve the success rate.

	Duration	Dates
Testing duration	15 days	09/01/2010 - 09/15/2010
Training durations	5 days	08/25/2010 - 08/29/2010
	10 days	08/20/2010 - 08/29/2010
	15 days	08/15/2010 - 08/29/2010
	20 days	08/10/2010 - 08/29/2010
	30 days	08/01/2010 - 08/30/2010

Table 3: Training and testing durations

KPI Type	KPI Label	β Coef.
Network delivery	RTP payload error	0.68
	Tuner fill	0.63
	Hole Too Large	0.61
	Decoder stall	0.42
	Bytes processed per sec	-0.32
Audio	Audio decoder errors	0.84
Video	Video DRM errors	0.73
	Video decoder errors	0.53
	Video frames decoded	-0.49
	Video data throughput	-0.49

Table 4: Significant KPIs for large δ (15-24 hrs)

KPI Type	KPI Label	β Coef.
Network delivery	Hole without session packets	0.60
	Tuner fill	0.57
	Bytes processed per sec	-0.34
	ECM parse errors	0.32
Audio	Audio decoder errors	1.03
	Audio samples dropped	0.84
	Audio crypto error	0.64
	Audio data dropped	0.55
	Audio DRM errors	0.34
Video	Video DRM errors	0.63

Table 5: Significant KPIs for small δ (3-9 hrs)

5. APPLICATION

In this section, we demonstrate the utility of Q-score by presenting three applications on it. First, we present a set of network KPIs that are closely related to user-perceived service quality. Second, we illustrate how much Q-score can predict user calls. Third, we show the possibility of intelligently dimensioning the call center workforce. In all applications, we successfully identify interesting results through online analysis of Q-score.

5.1 Identification of Significant KPIs

Today's commercial IPTV services support up to millions of user devices. If for every single device, few KPIs are monitored continuously, the measurement space can easily reach to the order of billions. In addition, time-lapse analysis in the diagnosis (as many diagnosis schemes employ) is required to be conducted on multiple data snapshots in short periods of time. Thus, in service assurance of a large-scale IPTV system, it is infeasible to blindly measure, collect, and analyze such large volume of diverse KPIs from the entire network. In this application, we discuss our experience on identification of a small number of significant KPIs with respect to user-perceived quality of experience.

Significant KPIs. In the generation of Q-score, we relate the network KPIs and user feedback by means of the factor β . β measures the relevance of significant KPIs by its magnitude. The analysis of the magnitude of β for different temporal aggregation levels indicates how KPIs correlate with user feedback. Tables 4 and 5 list top ten significant KPIs for relatively long history hours (15-24 hours) and short history hours (3-9 hours), respectively. Being regressed

KPI Type	KPI Label	β Coef.
Network delivery	Tuner fill	0.67
	Src unavailable received	0.5
	Hole without session packets	0.52
	ECM parse errors	0.35
	Bytes processed per sec	-0.33
Audio	Audio decoder errors	0.74
	Audio data dropped	0.57
	Audio crypto error	0.44
Video	Video DRM errors	0.68
	Video frames dropped	0.65

Table 6: Significant KPIs for multi-scale temporal aggregation (0-24 hrs)

with individual users’ feedback, the significant KPIs exhibit some commonality (shown in bold) as well as differences.

From the KPIs relevant to network delivery statistics, we observe that “tuner fill”, “hole without session packets”, “hole too large”, “bytes processed per sec” are particularly interesting KPIs. “Tuner fill” logs the number of packets lost by STBs before they are requested for TCP retransmission. The lost packets are supposed to be retransmitted by content distribution servers. Tuner fill counts can be related with video quality in that they indicate the condition of the delivery network and gives a sense of the average packet loss that would occur without any packet recovery scheme. A ‘hole’ represents a time interval greater than a given threshold (assumed to affect video quality) in which no video packets have been received. ‘Hole without session packets’ counts the number of such holes occurred during a STB’s viewing session (since the user’s last channel change). And ‘hole too large’ error is triggered when the hole size is larger than the maximum end-to-end delay of 150ms recommended by [2].

On the audio and video related KPIs, “decoder error” logs are general types of errors that occurred during the decoding of audio data. Decoder errors can occur due to various situations including, but not limited to, out-of-order data packet reception, audio buffer underrun or overrun, and packet loss. ‘DRM errors’ and ‘crypto error’ indicates errors caused by the video DRM decoder. This error can occur when encoder packets containing DRM keys are lost. In IPTV, every video program is encoded with DRM, and inability of decoding DRM blocks viewing of the programs. Thus, the occurrence of this error blocks TV viewing until new encoder keys are received regardless of receipt of the data packets. Lastly, the ‘video frames dropped’ error represents the number of video frames drops (below the normal frame rate of 29.97 frames per second) due to packet loss or decoder errors. When large frame drop occurs, viewers can notice choppy or skipy motions.

Observations. We observe an interesting finding by comparing significant KPIs of long-term event durations (*i.e.*, large δ) and short-term event durations (*i.e.*, small δ). The finding is that the former tend to have more video related KPIs as the most significant ones, whereas the latter has more KPIs related to audio. This relates with the relevance that audio has with respect to video in the user experience. Audio data is more susceptible to losses and errors than the video data. The reason is because the total volume of the data in audio is much less than that of the video, thus the impact of lost or delayed audio data is relatively greater than that of video data. Naturally, the viewers of the programs have less tolerance to audio issues than to video issues, and report about audio issues much earlier than video issues. The contrasting finding between long and short history hours has uncovered that, depending on the characteristics of the issues (*i.e.*, whether the issue is about audio or video), there are differences in urgency.

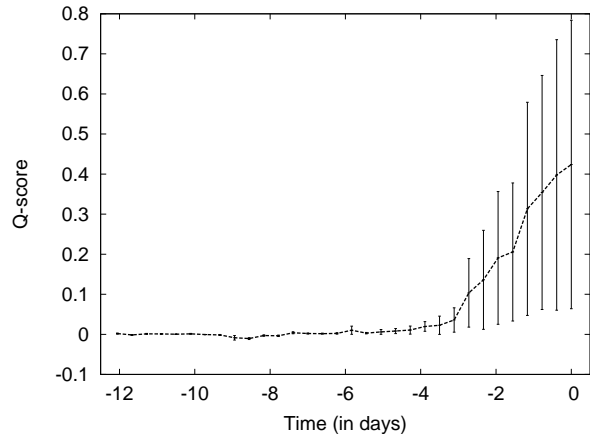


Figure 8: Growth pattern of Q-score

Another finding from the KPI analysis is drawn from multi-scale temporal aggregation. As shown in Table 6, by combining long-term and short-term event duration δ in regression, we observe both video and audio related issues appear as the most significant KPIs. This further confirms the effectiveness of letting the regression algorithm to choose important KPIs among multiple temporal aggregations.

Noticing that different KPIs have different degrees of relevancy to user feedback, we aim to guide monitoring of network KPIs by enlisting a small number of significant KPIs to user-perceived service quality. This way, forthcoming fine-grained network diagnosis can focus on the significant KPIs rather than analyzing excessive amount of KPIs.

5.2 Predicting Bad Quality of Experience

In order for Q-score to be useful for alerting services, it should have the capability to provide triggers well before users start to call. Thus, there is a need to study how much into the future we can infer customer calls using Q-score. To understand the feasible level of proactiveness in Q-score, we evaluated two characteristics: (i) the growth pattern of Q-score over time and (ii) stability of Q-score with a time gap between network events and user feedback.

Growth of Q-score Over Time. Figure 8 shows the growth pattern of Q-score for individual user IDs who filed trouble tickets. In the figure, we align the time by the trouble ticket filing time ($time = 0$) and observe how Q-score grows. The solid line represents the average value of the scores and the upper and lower tips of error bars represent one standard deviation plus and minus the average. From the graph, we observe that the increase of average Q-score is close to linear when it is greater than 0.05. The monotonic and gradual increase of Q-score suggests a possibility of using Q-score as a proactive trigger for alerting because (i) it keeps increasing once it becomes non-negligible level and (ii) its growth is not too abrupt. However, due to great variance among different users’ Q-scores, we cannot use Q-score of 0.05 as the significant value triggering forthcoming actions. Instead, we seek a more realistic lead time by conducting a further study on the stability of Q-score.

Feasible Level of Proactiveness. As aforementioned in Section 3.3, user feedback has indeterminate delay from the occurrences of network events. Here, we test the amount of lead time Q-score can provide before customer calls by measuring the accuracy loss as we increase the time gap between the occurrence times of network

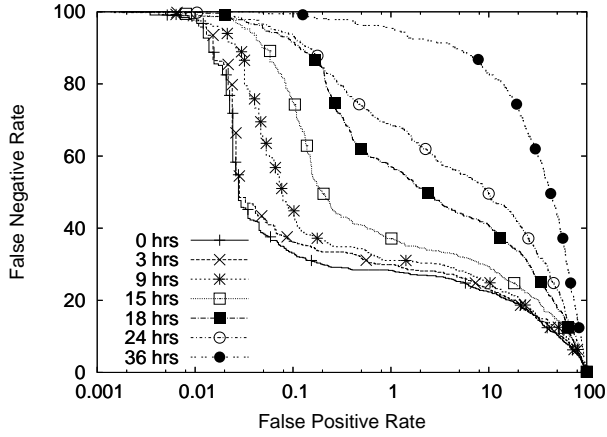


Figure 9: Comparison of accuracy of Q-scores with different skip intervals Δ

events b_i and user feedback c_i . The default time gap (or skipping interval) between $s_i + \delta$ and u_i is 0 hour because we set $u_i = s_i + \delta$ in Section 3.3. In this test, we add the skipping time gap Δ to the equation $u_i = s_i + \delta + \Delta$. By increasing Δ in the online monitoring step of Q-score generation, we test the regression for larger delays between b_i and c_i , in other words, we test for the stability of Q-score in proactive, early warning.

With various Δ ranging from 0 hours to 36 hours, Figure 9 exhibits FPR-FNR of learned β with different skipping times. As we increase Δ , the regression gets to rely on the user feedback of longer time after the occurrences of network events. And we observe that the FPR-FNR trade off gets worse as results. While the choice of lead time should mainly be left to the discretion of network administrators, we find 9 hours of lead time is at the feasible level, as observing 9 hours of skip interval preserves 0.1% of FPR only sacrificing 10% of FNR (*i.e.*, FNR is 30% when skip interval is 0 hours and 40% when skip interval is 9 hours).

5.3 Dimensioning Customer Care Workforce

If network problems occur locally to regional service areas rather than globally, an efficient management of field operators (*e.g.*, customer care representatives and repair men at customer premises) and servicing resources (*e.g.*, devices for fine-grained monitoring of network) would be to dynamically allocate them to challenging service regions than assigning static work areas. Thus, predicting the volume of forthcoming issues to a service region at a given time is beneficial in adaptively allocating workforce across service regions. In this application, we assess the possibility of pre-allocating a customer care workforce to potentially troubling service areas using Q-score. To begin, we first assess the volume of service quality issues per different spatial regions and see if the issues are contained locally or spread out globally.

Spatial Distribution of User Feedback. Figure 10 shows the spatial distribution of user feedback across different COs. The x -axis shows indexes of different COs, the z -axis shows temporal trend. The y -axis shows the amount of customer calls normalized by the peak value (*e.g.*, a value of 1 represents that the corresponding CO and time has the highest amount of calls shown in the figure)¹. At a given time, we observe that high user feedback is local to each

¹To protect proprietary information, we normalize some information in the results to the extent that the normalization does not obstruct interpretation of results

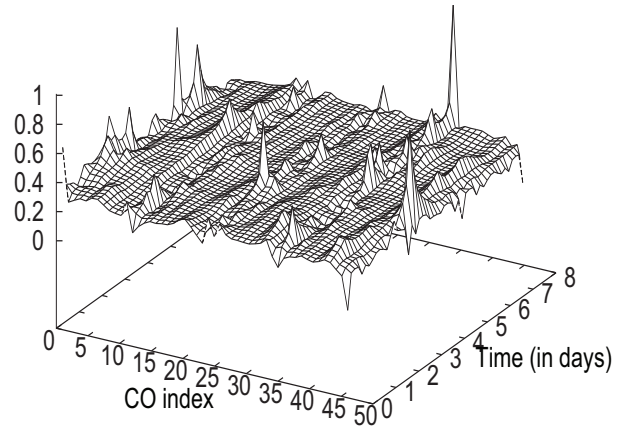


Figure 10: Normalized amount of users with customer reports over different spatial locations (COs) and times.

CO. And over time, the areas of high user feedback changes from one CO to another. From the fact that high feedback values generally being uncorrelated across time and CO (or space), we can affirm that the issues are temporal rather than permanent and local to an area rather than being global.

Leveraging Q-score for Dimensioning Workforce. Now that we have seen the possibility of dynamic resource allocation over different COs, we evaluate how closely Q-score follows user feedback in its magnitude when aggregated across individuals within each COs. Note that, to focus on its similarity to user feedback rate, we ignored the lead time of Q-score in this test. Figure 11 shows the trend of Q-score and user feedback aggregated per-CO. In doing so, Q-scores of individual user ID are first computed, and the scores corresponding to individuals within each CO are aggregated together to form per-CO Q-score. To compare three subfigures under the same scale factor, the plots are normalized by the peak customer call rate appearing in Figure 11(a), 22 hour time. Figure 11(a) shows the trend of per-CO Q-score and user feedback for a CO with relatively high customer feedback (*i.e.*, customer report rates). Over the course of 24 days, the percentage of users call the support center on the y -axis gets as high as 11%. Despite that there are some overestimations, the general trend of per-CO Q-score closely follows that of user feedback with Pearson's correlation coefficient $\mathcal{R} = 0.8797$. Figure 11(b) shows per-CO Q-score and user feedback for COs with moderately high customer feedback. We again see that the Q-score follows feedback whenever feedback increases over 2%. Here, $\mathcal{R} = 0.7478$ Figure 11(c) shows the same for a CO with few customer calls. Because there are only a small increase (2% of users calling) in the user feedback, Q-score remains at low level of 0.17% on average with $\mathcal{R} = 0.5011$. From the observations from three different COs with high, medium, and low level of feedback, we confirmed that Q-score, when aggregated across individuals within each CO, closely follows the trend of per-CO user feedback. Since Q-score is confirmed to have several hours of lead time before users begin to report, we can leverage Q-score in dimensioning the workforce and prioritizing resources to areas with more upcoming issues ahead of time.

6. RELATED WORK

In this section, we introduce related works on the two important components of networked service assurance: quality of experience assessment and network diagnosis.

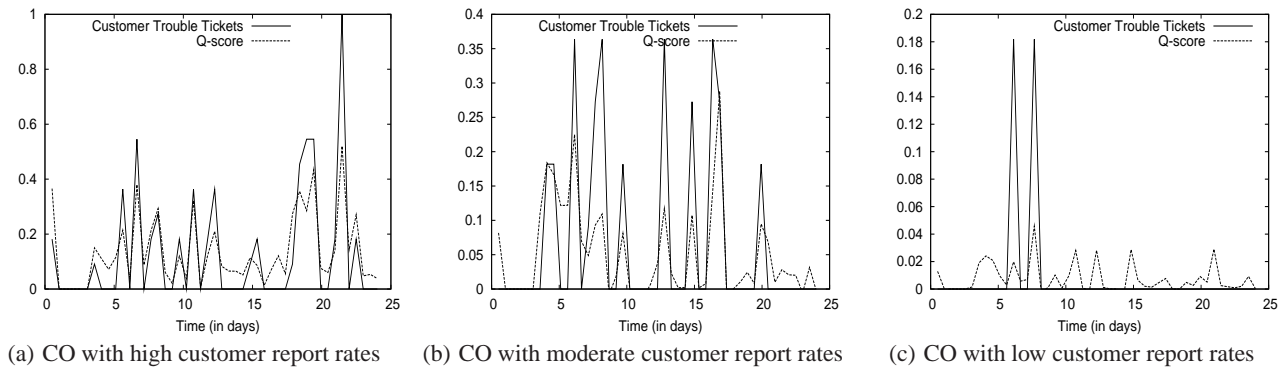


Figure 11: Trend of customer trouble tickets and Q-score per CO.

6.1 Quality of Experience Assessment

Controlled Performance Assessment. A traditional approach in the performance assessment of network devices is to use controlled lab environments. The code analysis, protocol analysis, testbed tests, and debugging done in such controlled environments serve to localize faults to each individual component of the system. [5, 6] apply principles of automated software fault diagnosis and model-based approaches. [21] outlines the methods of software and hardware verification using formal logic theorem provers. While theorem prover-based service assessment can be extensive and correct, the process of converting system operation to mathematical logic and inventing theorems thereafter restrict its application to a few specialized software and hardware systems such as that of the CPU and its microcodes. For validating network protocols, there have been several proposals [8, 12] based on simulation and system modeling using finite state machine, traffic modeling, and queuing theory.

Controlled testing has been extremely successful in detecting and preventing critical software bugs and hardware failures. Despite their best efforts, however, they are simply unable to replicate the immense scale and complexity of large operational systems and networks. Thus, there is always the risk of issues creeping into operational settings when they are missed in controlled environments. In this paper, we focus on a data-oriented mining approach that analyzes the data collected from an operational network. We believe a combination of data mining, lab reproduction, and software/hardware analysis is required to correctly identify anomalous service quality.

Video Quality Assessment. Subjective evaluation is the most reliable way of assessing the quality of an image or video, as humans are the final judges of the video quality in great part of the video related applications. The mean opinion score (MOS) [1] is a subjective quality measurement used in subjective tests which has been regarded as the most reliable assessment for video. However, subjective video assessment method is very inconvenient, expensive and slow. Thus there is a field of research dedicated to the design and development of objective quality metrics. Ongoing studies are both on standardizing the subjective measurement of video quality [23] and on developing objective video quality metrics that model and approximate the quality [3].

There are also video quality measurement studies in the context of networked systems [4]. The work includes discussions on the metrics of video quality measurable from various parts of a network. [17] studies the viewers' perception of video quality under packet loss-induced video impairments. [27, 28] proposes a loss-

distortion model based PSNR metric applied to video quality monitoring. Recently, ITU and other standardizing organizations began to roll out video quality measures such as [22]. Besides the lack of consensus in arriving at a single formula, video quality metrics may not be readily usable in the context of network service quality assessment as they require fine-grained measurements of per flow network traffic which current services dismiss due to the costs of measurement and data collection. While our method uses the customer trouble ticket as a proxy for user feedback, the concept of our methodology is open to employing a variety of video quality metrics as the measure of user experience.

6.2 Reactive Performance Diagnosis

Bayesian network and graph analysis are among the most widely used techniques in the diagnosis of network performance issues and troubleshooting [7, 10, 14, 16, 26, 29]. Kompella *et al.* [16] model the fault diagnosis problem using a bipartite graph and uses risk modeling to map high-level failure notifications into lower-layer root causes. WISE [29] presents a what-if analysis tool to estimate the effects of network configuration changes on service response times. Recent systems [15, 25] have used information available to the OS to identify service quality issues using the dependency structure between components.

[18–20] have shown the importance of focusing on recurring and persistent events and enabling the detection and troubleshooting of network behavior modes that have been previously flown under the operations radar. NICE [20] focuses on detecting and troubleshooting undesirable chronic network conditions using statistical correlations. Giza [18] applies multi-resolution techniques to localize regions in IPTV network with significant problems and l_1 -norm minimization to discover causality between event-series. Mercury [19] focuses on detecting the long-term, persistent impact of network upgrades on key performance metrics via statistical mining. A work on proactive prediction of service issues on access network [13] focuses on capturing changes over long-term (*e.g.*, weeks and months) and conduct prediction. The main difference between the above methods and ours is in the proactiveness of assessing service quality of experience (QoE). The reactive performance diagnosis works mostly focus on network problems but not on service quality of experience. We believe, Q-score is the first work in using the network performance indicators to proactively construct the quality of experience scores for large services. By capturing the quality of experience for users in a timely and scalable fashion, Q-score offers the operators with rapid notification of user-perceived issues and a lead time of several hours before customer reports.

7. CONCLUSION

In this paper, we develop Q-score, a novel framework for proactive assessment of user perceived service quality in a large operational IPTV network. By associating coarse-grained network KPIs with imperfect user feedback, Q-score generates a single score that represents user-perceived quality of experience (QoE). Accuracy analysis of Q-score reveals that it is able to predict 60% of service problems reported by customers with only 0.1% of false positive rate. Applying Q-score to various application scenarios, we have: (i) identified a set of KPIs most relevant to user-perceived quality of experience; (ii) quantified how early it can alert bad quality of experience; (iii) observed the possibility to pre-allocate the customer care workforce to potentially affected service areas.

As an improvement of our work, we consider the following two methods aimed at increasing the successful prediction rate. First, to filter out more noise from user feedback, we plan to investigate the trouble tickets that fell into false negatives. Collaborating with video experts, we will conduct simulation based controlled test-bed experiments in conjunction with our current operational data-driven approach. Second, to make Q-score to be more resilient to incompleteness of user feedback, we will further improve user grouping methods. In doing so, we plan on applying end-user clustering techniques in relation to user-perceived QoE.

There are many other network services that are sensitive to service quality that lack objective measures of user-perceived quality of experience. Our future work includes applying the proactive service quality assessment beyond the specific context of IPTV networks. For example, we plan to apply Q-score to VoIP and mobile networks so that operation teams can predict call drops and voice quality degradation without having to wait for customers to report them.

8. REFERENCES

- [1] Methods for subjective determination of transmission quality. *ITU-T Rec. P.800*, 1998.
- [2] One-way transmission time. *ITU-T Rec. G.114*, 2003.
- [3] Objective perceptual multimedia video quality measurement in the presence of a full reference. *ITU-T Rec. J.247*, 2008.
- [4] Perceptual audiovisual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference. *ITU-T Rec. J.246*, 2008.
- [5] R. Abreu, A. G. 0002, P. Zoetewij, and A. J. C. van Gemund. Automatic software fault localization using generic program invariants. In *SAC*, 2008.
- [6] R. Abreu, P. Zoetewij, and A. J. C. van Gemund. Spectrum-based multiple fault localization. In *ASE*, 2009.
- [7] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *Sigcomm*, 2007.
- [8] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in network simulation. *IEEE Computer*, 33(5), 2000.
- [9] S. Chatterjee and A. S. Hadi. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, Vol. 1:379–416, 1986.
- [10] I. Cohen, J. S. Chase, M. Goldszmidt, T. Kelly, and J. Symons. Correlating instrumentation data to system states: A building block for automated diagnosis and control. In *OSDI*, 2004.
- [11] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, Vol. 12, No. 1:55–67, 1970.
- [12] G. Holzmann. *Design and Validation of Computer Protocols*. Prentice-Hall, 1991.
- [13] Y. Jin, N. G. Duffield, A. Gerber, P. Haffner, S. Sen, and Z.-L. Zhang. Nevermind, the problem is already fixed: proactively detecting and troubleshooting customer dsl problems. In *CoNEXT*, page 7, 2010.
- [14] S. Kandula, D. Katabi, and J.-P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *MineNet*, 2005.
- [15] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed diagnosis in enterprise networks. In *ACM SIGCOMM*, 2009.
- [16] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *NSDI*, 2005.
- [17] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, and A. R. Reibman. A versatile model for packet loss visibility and its application in packet prioritization. *IEEE Transactions on Image Processing*, to appear, 2010.
- [18] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards automated performance diagnosis in a large IPTV network. In *ACM SIGCOMM*, 2009.
- [19] A. Mahimkar, H. H. Song, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and J. Emmons. Detecting the performance impact of upgrades in large operational networks. In *ACM SIGCOMM*, 2010.
- [20] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, and C. T. Ee. Troubleshooting chronic conditions in large IP networks. In *ACM CoNEXT*, 2008.
- [21] J. S. Moore and M. Kaufmann. Some key research problems in automated theorem proving for hardware and software verification. In *RACSAM*, 2004.
- [22] Perceptual evaluation of video quality. 2011. <http://www.pevq.org>.
- [23] M. Pinson and S. Wolf. Comparing subjective video quality testing methodologies. In *SPIE Video Communications and Image Processing Conference*, pages 8–11, 2003.
- [24] T. Qiu, J. Feng, Z. Ge, J. Wang, J. Xu, and J. Yates. Listen to me if you can: Tracking user experience of mobile network on social media. In *IMC*, 2010.
- [25] K. Shen, C. Stewart, C. Li, and X. Li. Reference-driven performance anomaly identification. In *SIGMETRICS*, 2009.
- [26] M. Steinder and A. Sethi. Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms. In *Infocom*, 2002.
- [27] S. Tao, J. Apostolopoulos, and R. Guérin. Real-time monitoring of video quality in ip networks. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '05, pages 129–134, New York, NY, USA, 2005. ACM.
- [28] S. Tao, J. Apostolopoulos, and R. Guérin. Real-time monitoring of video quality in ip networks. *IEEE/ACM Trans. Netw.*, 16:1052–1065, October 2008.
- [29] M. Tariq, A. Zeitoun, V. Valancius, N. Feamster, and M. Ammar. Answering what-if deployment and configuration questions with WISE. In *SIGCOMM*, 2008.