Computer Networks 54 (2010) 3123–3143

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

End-to-end quality of service seen by applications: A statistical learning approach

Pablo Belzarena *, Laura Aspirot

Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

ARTICLE INFO

Article history:
Received 28 May 2009
Received in revised form 25 March 2010
Accepted 3 June 2010
Available online 9 June 2010
Responsible Editor: N. Agoulmine

Keywords:
End-to-end active measurements
Statistical learning
Nadaraya–Watson
Support Vector Machines
QoS

ABSTRACT

The focus of this work is on the estimation of quality of service (QoS) parameters seen by an application. Our proposal is based on end-to-end active measurements and statistical learning tools. We propose a methodology where the system is trained during short periods with application flows and probe packets bursts. We learn the relation between QoS parameters seen by the application and the state of the network path, which is inferred from the interarrival times of the probe packets bursts. We obtain a continuous non intrusive QoS monitoring methodology. We propose two different estimators of the network state and analyze them using Nadaraya–Watson estimator and Support Vector Machines (SVM) for regression. We compare these approaches and we show results obtained by simulations and by measures in operational networks.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Most multimedia services in packet switched networks have some end-to-end quality of service (QoS) constraints that should be enforced like delay, jitter and loss rate. However, end-to-end QoS parameters cannot in general be estimated from data obtained in each isolated router. Therefore, methodologies to perform end-to-end active measurements and estimations have been developed during the last 10 years. Examples of such methodologies can be found in [1–12].

Some Internet service operators are offering now one or more "premium" services like video on demand, high quality video conferences, high definition IPTV, telematic services with real time requirements, etc. However, the rate at which these services have grown is smaller than initially expected. One of the main reasons behind this slowness is probably the difficulties that exist in the current Internet architecture to guarantee end-to-end QoS. The different proposals developed during the last 15 years in order to assure QoS (like IntServ, DiffServ, etc.) were not broadly deployed by the operators.

These difficulties are only exacerbated when the service provider offers a service spanning multiple domains. In this case, nodes of the path are under the administration of different network operators. Yet another aspect that further complicates the problem is the increasing heterogeneity of the access technologies (xdsl, cablemodem, wifi, wimax, 2G, 3G, mesh networks, etc.).

In the previous context, an important issue in order to control the end-to-end QoS is admission control. In a premium services network, an admission control mechanism based on the end-to-end performance helps the operator to control the end-to-end QoS. This constitutes one of the main motivations of this work.

An end-to-end admission control tool involves many different tasks. In this work we focus specifically in how to monitor the network in order to predict the end-to-end QoS seen by a premium service. With this information an admission control tool can decide whether it accepts a new service request or not. Although this problem is our main motivation, the tools proposed in this work can be

* Corresponding author. Tel.: +598 99687330.
E-mail addresses: belza@fing.edu.uy, pablobelzarena@gmail.com (P. Belzarena), aspirot@fing.edu.uy (L. Aspirot).

1389-1286/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2010.06.004

applied to many other operation and management problems; e.g. continuous monitoring of a Service Level Agreement (SLA).

A possible measurement technique for such monitoring tool is to send the application traffic (a video for example) and to measure then its QoS parameters at the receiver. However, in many cases these application flows have bandwidth requirements that are not negligible compared with links capacity. This technique could overload a congested link, degrading the QoS perceived by clients using the system. In addition, it may only be used then if the measurements are infrequent, and sporadic QoS degradations are tolerated. However, this is clearly not the case if the operator requires a permanent or frequent network monitoring.

In order to avoid the possibility of being themselves the cause of congestion, some measuring techniques estimate the QoS parameters seen by an application using light probe packets. However, these methodologies do not consider the particular characteristics of the application. In this sense, they implicitly assume, for instance, that the delay of a specific application can be approximated by the probe packets delay. This assumption is naturally not always true since QoS parameters depend on the statistical behavior of each type of traffic. Therefore, in many cases, this kind of estimation yields inaccurate results.

We propose a methodology that is an intermediate point between both approaches (to send a multimedia flow during long periods or to send light probe packets during short periods) and provides an accurate estimation of QoS parameters seen by an application without overloading the network during long periods.

The basic idea is to learn the relation between the probe packets interarrival times statistic and the QoS parameters seen by an application. We will assume that the former characterizes the state of the network. Once the relation has been learned, we may predict the QoS parameters just by sending light probe packets.

More formally, we consider the regression model

$$Y = \Phi(X) + \varepsilon,$$

where X, Y and ε are random variables. The random variable X is an estimation of the state of the network, the response Y is the QoS parameter seen by the application (delay, jitter, loss rate, etc.) and ε is a centered random variable which represents possible errors (e.g. in modelling, measurements, etc.) where ε and X are assumed independent.

The previous formulation evidences two problems. First, it is necessary to find an accurate estimation of the state of the network (the variable X). Second, it is necessary to estimate the function Φ. We estimate this function learning Φ from samples of the random variables Y and X.

Concerning the estimation of the state of the network, this work presents two contributions. The first one is a functional approach, where the estimation considers the empirical distribution function of probe packets interarrival times. This allows us to take into account other features that we cannot capture for example with the mean of interarrival times. However, this approach presents some drawbacks, in particular that it does not take into account time correlations. The second contribution addresses this point, by considering another estimator of the state of the network that captures information about time correlations. This estimator is presented in detail in Section 5. As we shall see, it obtains very good results and in some cases it is necessary to consider only a very small subset of parameters related with this estimator to have accurate estimations of the QoS.

On the other hand in order to estimate the function Φ we propose a statistical learning approach based on two different tools: Nadaraya–Watson estimator and Support Vector Machines (SVM). Nadaraya–Watson, a method which was originally devised for real data [13], is used in this work mainly for functional regression [14]. In particular we will use SVM in its regression variant, called Support Vector Regression (SVR). SVR has been extensively used for many applications since the nineties [15,16]. However, networking researchers started to apply SVR only a few years ago [17–19]. The nonparametric approach considered in this work differs from others in the literature, as will be discussed in Section 8). Our main contribution in this aspect consists in analyzing the use of these two estimators in the case of nonstationary data. In particular, we provide theoretical insight for applying a functional Nadaraya–Watson estimator in a nonstationary context. Moreover, we study the impact of nonstationary data when applying SVM techniques. In this case we present an implementation of SVM that leads to accurate estimates even in the presence of nonstationarity.

We organize this paper in the following way. In Section 2 we detail our approach to monitor the QoS parameters seen by an application. In Section 3 there is a brief introduction to the statistical learning tools used in this work: Nadaraya–Watson and Support Vector Machines. In Section 4 we characterize the state of the network by the empirical distribution of the probe packets interarrival times and the QoS estimation is based on functional Nadaraya–Watson. We analyze some preliminary experimental results for video applications. In Section 5 we propose an alternative estimator of the state of the network, one which is related with the length of the queues. We also discuss the advantages of this estimator when applying SVM tools. In Section 6 we analyze the impact of video characteristics in video QoS estimations. In Section 7 we show results from experiments in three different operational networks. In Section 8 we analyze other works that estimates QoS seen by applications. Finally, in Section 9, we discuss the main conclusions of this work.

2. Problem formulation and proposed solution

We first consider the case of a path with a single-link. The multi-link case is discussed later. We assume that the cross traffic, the link capacity and the buffer size are unknown. The QoS parameter seen by the application is called Y and it is a function of the link and traffic characteristics:

$$Y = F(X_t, V_t, C, B),$$

where X_t is the cross traffic stochastic process, V_t is the stochastic process corresponding to the application traffic, C is the link capacity and B is the buffer size.

Download English Version:

https://daneshyari.com/en/article/453077

Download Persian Version:

https://daneshyari.com/article/453077

Daneshyari.com