On the QoS Estimation in an OpenFlow Network: The Packet Loss Case

Marco Cello, Mario Marchese, *Senior Member, IEEE*, and Maurizio Mongelli, *Senior Member, IEEE*

***Abstract*—OpenFlow specification for SDN networks leaves to external entities the control of the queues inside the switches of the network. SDN typically re-routes flows on less congested paths before they experience QoS violation. This letter outlines an original solution to predict the effect of the re-routing operation by focusing on the packet loss metric. Since the approach is based on collected real-time statistics, it can be easily generalized to other QoS metrics and measurement methodologies.**

***Index Terms*—SDN, OpenFlow, packet loss, neural estimation.**

I. Introduction

In OpenFlow-enabled switches, queue configuration (service rate, buffer size) takes place outside the OpenFlow protocol [1]. Flows are routed or re-routed on different paths according to the current level of congestion. In this context, we derive a methodology to predict in advance the experienced QoS as a result of the re-routing operation. The QoS metrics considered is the packet loss. Since the approach is based on collected statistics without a-priori assumptions on traffic and working conditions, it is applicable to any QoS metric whose time evolution is monitored by the system. The mechanism is also independent to both the way the QoS is monitored and the target of the prediction, which is the QoS change for congestion. This is a significant advantage because there are several distinct motivations for QoS degradation (see, e.g., [2] for an informal discussion about troubleshooting of loss monitoring via the ping tool). Nothing prevents to let the prediction discriminate separate events leading to the same event of QoS degradation (see, e.g., [3]); this however is left open for future research.

The remainder of this letter is structured as follows. Section II shows the motivations that lead to consider the re-routing concept and the QoS estimation problem in OpenFlow. Section III describes the QoS estimation framework. The performance analysis is presented in IV and the conclusions are drawn in Section V.

II. Technological Problem

Current OpenFlow (OF) specification [1], however, gives the responsibility of rates configuration to *external protocols*. Information exchange between QoS management plane and OF is an implementation-specific choice. *"Queue configuration takes place outside the OpenFlow protocol, either through a command line tool or through an external dedicated configuration protocol"*. (Section 7.3.5.8, [1]).

In this perspective, we exploit the quantities returned by OF as described in the following section in order to build a set of features which drive QoS forecast before a re-routing operation. We state the estimation problem on a single traffic buffer. In virtue of the centralization of all the measures at the SDN controller, the end-to-end QoS may be derived by repeating instances of the same problem along the path of the new flow.

III. QoS Forecast

A. Forecast Model

The proposed methodology is derived by formulating an estimation problem of the QoS in correspondence of the current and next states of a traffic queue, i.e., before and after the addition of new flows. A sequence of estimation steps $k = 1, 2, \ldots,$ is defined such that, on the basis of feedback acquired during the system evolution, an estimation law $f(\cdot)$ gives indication about the QoS achieved when a new flow (or a group of flows) enters the buffer. Such estimation law, or *predictor*, represents a function that performs a mathematical mapping between the acquired feedback and the future QoS[1]. The feedback consists of the information available about the current state of the network and the basic statistics of the new flow. The buffer loss in terms of percentage is the reference QoS metric.

We assume a flow may enter at any time. Two consecutive time horizons $[k-1, k]$ and $[k, k+1]$ are defined in correspondence of the time k when the new flow enters the buffer. The size of the horizons is T. The quantities $q(k)$ and $q(k+1)$ denote the QoS metric over the respective horizons $[k-1, k]$ and $[k, k+1]$. If ν new flows enter at time k, they increase the number of active sources from $N(k)$ in $[k-1, k]$, to $N(k+1) = N(k) + \nu$ in $[k, k+1]$. A forecast of $q(k+1)$ at time k is required to understand if the new flows may be disruptive for the QoS or not. The estimation operation is driven by the estimation law with: $q(k+1) = f(I(k))$.

The quantity $I(k)$ is a finite-dimensional information vector collecting the observations of the features of interest acquired in the time interval $[k-1, k]$. $I(\cdot)$ should contain only basic statistics collected from OF, such as the mean m and standard deviation σ of the overall input flow rate of the buffer. As formalized above for the QoS, m and σ are computed both for

Manuscript received August 7, 2015; accepted December 30, 2015. Date of publication January 11, 2016; date of current version March 8, 2016. The associate editor coordinating the review of this paper and approving it for publication was B. Rong.

M. Cello and M. Marchese are with the DITEN Department, University of Genoa, Genova 16145, Italy (e-mail: marco.cello@unige.it; mario.marchese@unige.it).

M. Mongelli is with the Institute of Electronics, Computer and Telecommunication Engineering, National Research Council of Italy, Genova 16149, Italy (e-mail: maurizio.mongelli@ieiit.cnr.it).

Digital Object Identifier 10.1109/LCOMM.2016.2516537

[1] In statistics and in machine learning, a predictor function is a linear/non-linear function (linear/non-linear combination) of a set of coefficients and explanatory variables (independent variables, the information vector defined later on), whose values are used to forecast the outcome of a dependent variable (the QoS, in our case)

the $[k-1,k]$ interval: $m(k)$, $\sigma(k)$ and the successive interval $[k,k+1]$: $m(k+1)$, $\sigma(k+1)$. $m(k)$ and $\sigma(k)$ are got by OF directly from measurements of the buffer[2]. Since $m(k+1)$ and $\sigma(k+1)$ correspond to the presence of the new flows before they really enter the buffer, basic statistics are set as follows: $m(k+1)=m(k)+m'$, $\sigma(k+1)=\sigma(k)+\sigma'$[3], m' and σ' being the statistics of the new flows. In case of re-routing, those statistics are returned from OF because they are already present in the network. If the flows come from outside of the network (routing case), the statistics are supposed to be known in advance.

$I(\cdot)$ may also contain information about the statistical properties of the traffic sources, denoted in compact form with the $\boldsymbol{p}$ vector[4]. $I(k)$ thus can assume the following generic form (the precise definition of the operative used $I(k)$ will be given in subsection III-D):

$$\begin{aligned}\boldsymbol{I}(k) = [&T, \theta(k), B_{Max}(k), \boldsymbol{p}(k),\\ &N(k), m(k), \sigma(k), q(k), B(k),\\ &N(k+1), m(k+1), \sigma(k+1)]\end{aligned} \tag{1}$$

where $\theta(k)$ is the queue rate assigned during the $[k-1,k]$ horizon (outside of OF control), B and B_{Max} are the current and maximum buffer sizes, respectively. The first three parameters ($\theta(k)$, $B_{Max}(k)$, $\boldsymbol{p}(k)$) define the behavior of the other ones, whose fluctuations lie over shorter time scales. $\theta(k)$, $B_{Max}(k)$, $\boldsymbol{p}(k)$ are however included in the information vector in order to evaluate their impact on the estimation error. The presence of $\boldsymbol{p}$ in the information vector is very important: the knowledge of a precise description of the sources has an impact on the estimation performance, as evidenced later. The use of T in $\boldsymbol{I}(\cdot)$ is motivated as follows. Since the tracking of performance over fixed time horizons may be noisy in dependence of T, we pursue a self-adaptive mechanism to variable T, thus avoiding the need of building different $f(\cdot)$ with respect to different T. In other words, the mechanism learns the level of noise corresponding to the specific T and correct it. This is a significant advantage with respect to other traditional estimation approaches, like Kalman or ARMA, which does not take into account T explicitly.

B. Neural Approximation of the Estimation Law

The estimation law $f(\cdot)$ is derived by neural approximation. This is a common approach when the unknown prediction law may contain non linearities and the noise may be non-stationary and non-Gaussian (see, e.g., [4], [5]). A neural network (NN) is defined with the same input $I(\cdot)$ and with weights $\boldsymbol{\omega}$: $\hat{f}(I(\cdot),\boldsymbol{\omega})$. A regular neural training problem is then derived from a database containing $N_{\hat{f}}$ couples $\{\boldsymbol{I}(\varsigma); q(\varsigma+1)\}$ for each sample ς, $\varsigma=1,\ldots,N_{\hat{f}}$ (no matter how the samples are obtained, i.e., simulatively or by sampling the real system). The mentioned training consists of finding the weights assignment $\boldsymbol{\omega}^*$ so that:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}} J(\boldsymbol{\omega});\ J(\boldsymbol{\omega}) = \sum_{\varsigma=1}^{N_{\hat{f}}}[q(\varsigma+1)-\hat{f}(\boldsymbol{I}(\varsigma),\boldsymbol{\omega}^*)]^2. \tag{2}$$

The problem (2) is a regular neural regression scheme that tunes the output of the NN in order to approximate the collected values of $q(\varsigma+1)$ as a function of the information vectors $\boldsymbol{I}(\varsigma)$, $\varsigma=1,\ldots,N_{\hat{f}}$. Operatively, once (2) is solved, the resulting NN may be used to infer new values of q according to new values of $\boldsymbol{I}$.

C. Neural Approximation Driven by Effective Bandwidth

QoS control belongs to the well known effective bandwidth (EfB) framework [6], for which several solutions exist in the literature. EfB formulas typically return closed-form expressions of QoS metrics as a function of the service rate and of other traffic parameters. EfB, however, typically exploits the knowledge of traffic descriptors, which are not included in the information carried by OF. EfB techniques may be hardly applied without the explicit knowledge of the traffic descriptors ($\boldsymbol{p}$). For the loss metric and through basic statistics (mean and standard deviation), the well-known EfB formula [7] may be applied:

$$\theta = m + \epsilon_1\cdot\sigma;\ \epsilon_1 = \sqrt{-2ln(l_{EfB}) - ln(2\pi)} \tag{3}$$

l_{EfB} being the EfB loss under the assigned rate θ. An alternative to the use of the information vector may be derived as follows. The first step is inverting (3):

$$l_{EfB} = e^{-\frac{\epsilon_2}{2}};\ \epsilon_2 = \left(\frac{\theta-m}{\sigma}\right)^2 + ln(2\pi) \tag{4}$$

Since (3) and (4) assume a Gaussian distribution of the input bit rate and do not exploit the buffer size, the q forecast through l_{EfB} may be inefficient. Namely, the real loss l may be not perfectly aligned with l_{EfB}. Formula (4) may be however exploited to introduce additional information into $I(k)$. The values of the parameter $\delta = l_{EfB} - l$ at time k and $k+1$ are added into (1) and the target of the regression problem (2), $q(k+1)$, is replaced with $\delta(k+1)$. As a result of this, the NN is trained to predict the error introduced by the EfB formula at time $k+1$, under the current traffic change, in place of directly addressing the QoS forecast $q(k+1)$. The subsequent derivation of the loss forecast is straightforward from the predicted δ. The prediction of the error over the heuristic forecast (δ) is preferable than directly addressing the neural approximation of the performance metric (q). The rationale behind this relies on the fact that the values of δ may have a smaller variance and less discontinuities with respect to q.

[2]The SDN controller makes available the following quantities: 1) the estimated rate per flow that is obtained by subtracting the number of transmitted bytes, of two consecutive statistics reply messages and divided it by the polling interval; 2) the number of flows belonging to a specific queue.

[3]The summation operation exactly foresees the average and standard deviation of the new aggregate in $[k,k+1]$ only in the presence of a Gaussian behavior of the sources. The estimation law is thus also responsible for correcting the error introduced by the presence of non-Gaussian sources.

[4]For example, $\boldsymbol{p}$ may include information of: N_{Max}, the maximum number of traffic sources, Bp and β, the peak bandwidth and the burstiness of the sources, respectively (the burstiness is the ratio between the peak and the average bandwidth of the sources). The results include a setting related to datacenter environments.

D. Information Vectors

The following structures of the information vector are considered.

$$\boldsymbol{I}_{min}(k) = [\theta(k), B_{Max}(k), m(k), \sigma(k), q(k), m(k+1), \sigma(k+1)] \tag{5}$$

defining the minimum information available to infer the new $q(k+1)$ after the addition of the new flows.

$$\boldsymbol{I}_{EfB}(k) = [\boldsymbol{I}_{min}(k), l_{EfB}(k), \delta(k), l_{EfB}(k+1)] \tag{6}$$

The EfB quantities defined in the previous subsection are added to $\boldsymbol{I}_{min}(\cdot)$ to obtain $\boldsymbol{I}_{EfB}(\cdot)$. The inference is first on $\delta(k+1)$ and then on $q(k+1)$ through $l_{EfB}(k+1)$.

$$\boldsymbol{I}_{full}(k) = [\boldsymbol{I}_{EfB}(k), \boldsymbol{p}(k), N(k), N(k+1)] \tag{7}$$

All available information is exploited. The inference is again first on $\delta(k+1)$ because EfB quantities are included in $\boldsymbol{I}_{full}(\cdot)$. The $\boldsymbol{p}$ vector is assumed constant in $[k-1, k+1]$ and only the sample at time k is used. However, nothing prevents to consider the addition of flows having new traffic descriptors such that $\boldsymbol{p}(k+1) \neq \boldsymbol{p}(k)$. Despite the $N(k)$, $N(k+1)$ quantities may be known in OF, they have been excluded from $\boldsymbol{I}_{min}(\cdot)$ and $\boldsymbol{I}_{EfB}(\cdot)$ to stress the working conditions. It is finally worth noting that both $\boldsymbol{I}_{min}(\cdot)$ and $\boldsymbol{I}_{EfB}(\cdot)$ would be really applicable in a OF network, while $\boldsymbol{I}_{full}(\cdot)$ is considered for performance comparison only.

IV. Performance Analysis

A. Traffic Traces

1) Datacenter: The synthetic generation of data traffic in a datacenter (DC) is considered because SDN is often used for DCs. We generate an ON-OFF stochastic process in which ON and OFF durations along with the inter-arrival duration within ON periods are random variables that follow the Lognormal distribution. We select the distributions' parameters (i.e., *location* μ and *scale* ξ) that lead to a cumulative distribution function (CDF) similar[5] to the CDFs of the experimental traces shown in Figures 6–8 of [8]. The resulting setting is: $\mu_{ON} = 1, \xi_{ON} = 0.5$, $\mu_{OFF} = 0.5, \xi_{OFF} = 2$, and $\mu_{inter} = 0.1, \xi_{inter} = 0.5$. We verified that different subsets of ON-OFF flows following those distributions have similar statistical characteristics [6].

2) Multimedia: The ON-OFF traffic model of multimedia applications [9] is considered as well. Each source is an ON-OFF process. Burst and silence durations are exponentially distributed. For both the traffic models, the traffic buffer is simulated using an ad-hoc C++ simulator. In virtue of the independence of the simulations operated to build the database of the samples $\{\boldsymbol{I}(\varsigma); q(\varsigma+1)\}$ (outlined in subsection III-B), the processing time can be significantly reduced by using multiple CPUs.

B. Results

1. Datacenter: Training: The database is built in 6.5 hours over an IntelCore i7-3630QM@2.4GHz, over which 6 simulation threads may run in parallel. The database contains 30000 samples of $\{\boldsymbol{I}(\varsigma); q(\varsigma+1)\}$. The packet size is fixed to 1500 Byte and $N_{Max} \in [1, 100]$ (the maximum number of connections), $B_{Max} \in [100, 1100]$ packets, the μ_{ON} and μ_{inter} parameters of the Lognormal distributions are extracted in [0.1, 2] and [0.1, 1], respectively. The vector of traffic descriptors is therefore $\boldsymbol{p} = [\mu_{ON}, \mu_{inter}]$.

The amount of new flows entering the buffer is set as follows. At the beginning, the number of active sources is $N = N_{Max}/2 + \nu, \nu \in [1, N_{Max}/3]$. The number of sources abruptly added to the buffer is $N_{Max} - N$. This implies that from the 20% of N_{Max} up to the 50% of N_{Max} may constitute the amount of new flows.

The ranges considered for the variables and the considered proportion of new flows imply a high variability of system conditions, which may considerably stress the prediction model. The service rate of the buffer is set according to $\theta = 2 \cdot N_{Max}$ Mbps. The corresponding rate allocations lie in the range [2, 198] Mbps, with an average of 100 Mbps. The measured average target (i.e., the loss after the addition of new flows) over the training set is 6%; the average loss before the addition of new flows is 1%.

A neural network with 20 hidden neural units with hyperbolic tangent activation function is used for the regression scheme (2) under all the considered information sets: $\boldsymbol{I}_{min}$, $\boldsymbol{I}_{EfB}$ and $\boldsymbol{I}_{full}$. The training is addressed by the Neural Network Toolbox in Matlab (version R2014b) by using the Levenberg-Marquardt algorithm. The database driving the training phase is divided in 70% of samples for training and 15% for both validation and test. To avoid overfitting, the minimum number of hidden neural units is empirically found in correspondence of the minimum J, defined in (2), over the test set. The corresponding computational time is approximately 60 s on the same computer if $\boldsymbol{I}_{min}$ is used. The other information vectors lead to sensibly lower computational times because more information simplifies the training problem.

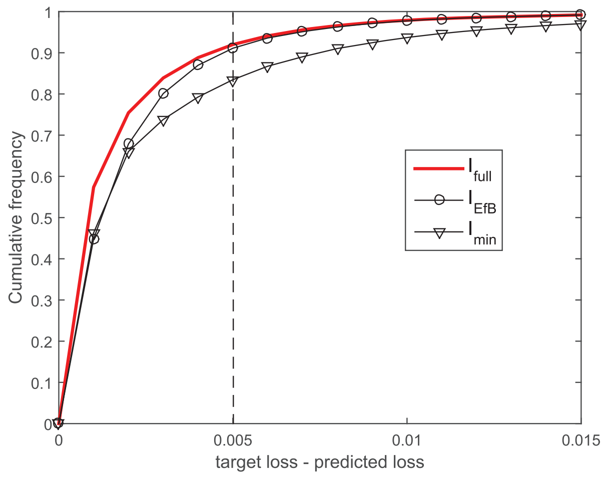
Test. After training, 10000 new independent repetitions are generated by extracting the system parameters from a set of random uniform distributions in the ranges outlined above. Both in training and test, the reference period T for sampling the information vector and evaluating the performance lies in [10, 240] s.

Fig. 1-(a) shows the cumulative frequency[7] of the difference between target (i.e., the real loss, unknown to the predictor) and predicted loss. The impact of the change of the information set is appreciable. For example, as highlighted by the dashed vertical line in the figure, the difference is lower than $5 \cdot 10^{-3}$ (i.e., it is a order of magnitude lower than the average target) with probability 0.91 under $\boldsymbol{I}_{full}$ and $\boldsymbol{I}_{EfB}$ and with probability

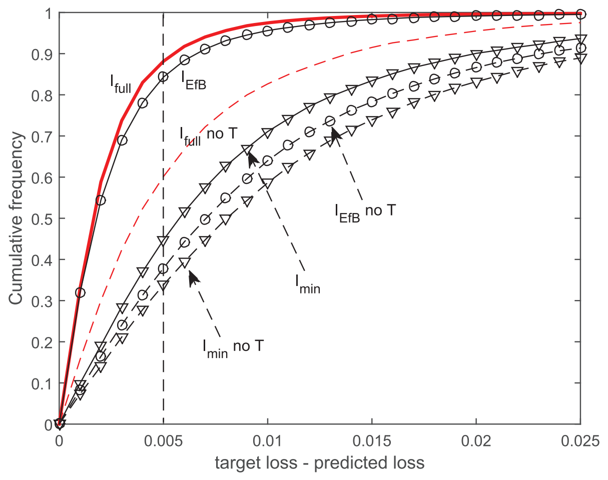
[5]Given two cumulative distribution functions $\mathrm{CDF}_1(x)$, $\mathrm{CDF}_2(x)$, the following distance metric should be lower than a given threshold: $\int_0^{+\infty} \|\mathrm{CDF}_1(x) - \mathrm{CDF}_2(x)\| dx$.

[6]This was validated through q-q plot analysis by comparing different subsets with $N \in [1, 100]$ and with the ranges on μ_{ON} and μ_{inter} used in the simulations.

[7]It is the relative frequency of events such that the real loss and the estimated loss differ by a value $\leq x$, where x lies in the abscissa of the presented figures.



(a) Datacenter.



(b) Multimedia.

Fig. 1. Cumulative frequency of the difference over the target loss (test set).

0.82 under $\boldsymbol{I}_{min}$. The gain of $\boldsymbol{I}_{EfB}$ over $\boldsymbol{I}_{min}$ is significant. $\boldsymbol{I}_{EfB}$ thus helps overcome the lack of knowledge of the traffic descriptors.

The impact of T in the information vectors is negligible: similar curves are obtained by repeating the calculations outlined in Fig. 1-(a) without the presence of T in the information vectors. The following correlation coefficients from Bivariate Statistics analysis (BSA) [10] are calculated with respect to all the variables and the target (i.e., $q(\varsigma + 1)$): Pearson, Spearman and Kendall tau. The corresponding values with T are around 0, thus denoting the two variables are independent. The highest correlations, i.e., values of the coefficients close to 1 or -1, are registered with $\boldsymbol{p}$ and $q(\varsigma)$.

2) Multimedia: The multimedia case follows a large range of working conditions as well: $Bp \in [10, 200]$ kbps (peak bandwidth of the single source), $N_{Max} \in [5, 100]$, $\beta \in [1, 6]$ (burstiness of the sources); $\boldsymbol{p} = [Bp, \beta]$. The buffer size is variable, too: $B_{Max} \in [100, 500]$. The service rate θ is set according to $\theta = Bp \cdot \beta^{-1} \cdot N_{Max}$. The corresponding allocations lie in the range [0.01, 14.87] Mbps, with an average of 1.9 Mbps. The other configurations are identical to the DC case, except for the NN, whose training time is slightly higher than 60 s with $\boldsymbol{I}_{min}$. The training database is built in less than 3 hours. The average loss over the test set is 5.6%. As evidenced by the vertical line in Fig. 1-(b), the difference is lower than $5 \cdot 10^{-3}$ with probability 0.87 under $\boldsymbol{I}_{full}$, with probability 0.845 under $\boldsymbol{I}_{EfB}$ and 0.44 under $\boldsymbol{I}_{min}$. A further validation deals with the prediction error with respect to a lower target loss probability of 1% (after the addition of the new flows). Fig. 1-(b) does not change qualitatively if we extract from the test set the losses lying in the range [0, 1%]. For example, the probability evidenced by the previous numerical example with $\boldsymbol{I}_{min}$ (i.e., 0.44) slightly decreases to 0.41. Fig. 1-(b) also evidences the impact of T: the curves using an information vector deprived of T are denoted with the "...; no T" addition and are dashed lines. The impact is relevant in this case, as also confirmed by BSA that shows high values of correlation of T with the target. Since BSA may be performed as a pre-processing step (before NN training), it may be a reliable indication on the potential advantage of including T in the information vectors.

V. Conclusion and Future Work

A neural predictor has been developed and tested for loss estimation in an OpenFlow environment. The reliability and generality of the approach lead to further investigations in terms of using other traffic features and QoS metrics.

We also created a real SDN testbed using Open vSwitch and D-ITG traffic generator. We would like to conduct extensive testbed measurements in order to investigate what peculiarities are not captured by simulations and are available through the platform, such as oscillations, transitions to different non-stationary states of traffic, congestion control or other streaming protocol (not easily applicable to a discrete event simulator) and so on.

References

[1] Open Networking Foundation. (2014, Dec. 19). *OpenFlow Switch Specification—Version 1.5.0* [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

[2] CodeWeblog.com. (2010). *Ping Troubleshooting* [Online]. Available: http://www.codeweblog.com/ping-packet-loss-always-causes-troubleshooting-ideas-solutions, accessed Nov. 28, 2015.

[3] A. Jayaraj, T. Venkatesh, and C. Murthy, "Loss classification in optical burst switching networks using machine learning techniques: Improving the performance of TCP," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 45–54, Aug. 2008.

[4] J. Rodrigues, A. Nogueira, and P. Salvador, "Improving the traffic prediction capability of neural networks using sliding window and multi-task learning mechanisms," in *Proc. 2nd Int. Conf. Evol. Internet (INTERNET)*, Sep. 2010, pp. 1–8.

[5] M. Mongelli and S. Scanzio, "Approximating optimal estimation of time offset synchronization with temperature variations," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 12, pp. 2872–2881, Dec. 2014.

[6] F. Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Notes Series*, vol. 4. Oxford: Clarendon, 1996, pp. 141–168.

[7] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 7, pp. 968–981, Sep. 1991.

[8] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in *Proc. 1st ACM Workshop Res. Enterp. Netw. (WREN'09)*, 2009, pp. 65–72 [Online]. Available: http://doi.acm.org/10.1145/1592681.1592692.

[9] D. McDysan, *QoS and Traffic Management in IP and ATM Networks*. New York, NY, USA: McGraw-Hill, 2000.

[10] J. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference: Revised and Expanded*, 4th ed. New York, NY, USA: Taylor & Francis, 2014 [Online]. Available: https://books.google.co.uk/books?id=kJbVO2G6VicC.