Poster: A Portfolio Theory Approach to Edge Traffic Engineering via Bayesian Networks[*]

Mary Hogan
Computer Science Department
Saint Louis University
mhogan26@slu.edu

Flavio Esposito
Computer Science Department
Saint Louis University
espositof@slu.edu

ABSTRACT

One of the main goals of mobile edge computing is to support new generation latency-sensitive networked applications. To manage such demanding applications, a fine-grained control of end-to-end paths is imperative. End-to-end delay estimation and forecast techniques were essential traffic engineering tools even before the mobile edge computing paradigm pushed the cloud closer to the end user. In this paper, we model the path selection problem for edge traffic engineering using a risk minimization technique inspired by portfolio theory in economics, and we use machine learning to estimate the risk of a path.

In particular, using real latency time series measurements, collected with and without the GENI testbed, we compare four short-horizon latency estimation techniques, commonly used by the finance community to estimate prices of volatile financial instruments. Our initial results suggest that a Bayesian Network approach may lead to good latency estimation performance and open a few research questions that we are currently exploring.

1 INTRODUCTION

Mobile Edge Computing is a fairly novel computing paradigm in which node processing and traffic engineering decisions may be offloaded from processes on mobile devices to the edge of the network. This approach has been shown to improve user experience by reducing the perceived latency, and is growing in popularity because of the Internet of Things (IoT) and the vast amount of data that sensors generate. It is inefficient to transmit all the data that a bundle of sensors creates to the cloud for processing and analysis; doing so requires a great deal of bandwidth and all the back-and-forth communication between the sensors and the cloud can negatively impact performance. Traffic engineering at the edge is critical not only to support future Internet of (medical) Things applications, but also to support delay-sensitive applications whose traffic spans across 5G customers or across several Points of Presence, *i.e.*, servers located at the edge of the network of application providers, see, *e.g.*, the Facebook's [?] or the Google's edge networks [?].

Most mobile edge computing applications orchestrate multiple processes that compete for network resources, causing data transmissions to become inefficient, or data paths to become congested or unreliable. This problem is exacerbated in mobile and challenged environments, where network connectivity is scarce, unavailable, and latency-sensitive applications need to pre-process real time data captured from IoT devices, *e.g.*, images captured by drones or by first responder devices in a natural or man-made disaster scenario. Existing forecast-based path management solutions for mobile and delay-sensitive applications are often tailored to specific protocols or applications [? ?], they focus on link bandwidth [?] or switch queue size estimation [?], they are not designed to dynamically steer traffic in a multi-path network [? ?], or they focus on maximizing performance of a single flow [? ?].

Our Contributions: In this paper, we present a path management solution that, leveraging results from portfolio theory and stochastic learning theory, helps edge traffic engineers identify end-to-end paths (routes) whose future estimated latency is minimized in a given horizon. In particular, leveraging a portfolio theory formulation [?], we first introduce an analytic model for the path selection problem in edge traffic engineering, capturing the risk-return factor associated with a network path choice and its latency. In our model, selecting a set of low-latency paths is equivalent to the problem of selecting a portfolio of assets, maximizing the expected return subject to a given level of (volatility) risk. Since our model captures the risk of a path with its predicted latency value, we need a valuable short-horizon latency estimation technique. To this end, we measured end-to-end latencies using the ICMP protocol over the GENI testbed [?], and compared the performance of (four) latency estimation techniques commonly used to predict future prices of a volatile financial instrument: a Bayesian network model [?], an autoregressive model, a moving average model, and an AutoRegressive Moving Average (ARMA) model. Our initial results show that the Bayesian network approach is the best latency (peak) predictor *i.e.*, it has the lowest relative error, as long as there are statistical dependencies among the latency measurements, and such measurements do not have latency variance too small.

The rest of the paper is organized as follows: In § ?? we present our model inspired by portfolio theory; in § ?? we describe how we applied the Bayesian network model to predict latencies and in § ?? we present our initial evaluation results. Finally, in § ?? we present the limitations of our study, the lesson learned and our ongoing and future work.

[*]This work has been supported by the NSF award CNS-1647084.

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-4916-1/17/10.
https://doi.org/10.1145/3117811.3131250

2 MODELING RISKY PATHS USING PORTFOLIO THEORY

Aside from its applications for financial asset allocation, we argue that portfolio theory [?] is a valuable resource allocation tool also for traffic engineering problems, especially in mobile edge computing, where latency is as crucial as stock prices. In this section we first give a background on portfolio theory, applied to a standard financial portfolio selection problem and then we describe how we use it to model risky paths for an edge traffic engineering problem.

Background: Portfolio Theory. Maximizing the return is undoubtedly the first goal of every investor. The second main characteristic of an investment is the level of perceived risk to obtain such return, compared to the average over the investment period. Portfolio theory [?] formalizes the problem of selecting a portfolio *i.e.*, the set of items (*e.g.*, financial instruments) that maximizes the expected return given some level of risk. The problem can be alternatively formulated as a risk minimization problem, given an expected value of return.

The classical portfolio problem considers n assets held over a period of time. Let us denote with z_i the dollar amount of asset i held throughout the investment period, at the price obtained at the beginning of the investment period. The simplest formulation does not consider obligations to buy assets at the end of the period, that would yield $z_i < 0$, so asset i always corresponds to $z_i > 0$; We let p_i denote the relative price change of asset i over the period, *i.e.*, its change in price over the period divided by its price at the beginning of the period. The overall dollar return on the portfolio is hence given by $r = p^T z$, where the optimization variable is the portfolio vector $z \in \mathbb{R}^n$. A wide variety of constraints on the portfolio can be considered. Let us consider $\mathbf{1}^T z = B$, that is, the total budget to be invested is B, which is often normalized to one.

Considering a stochastic model for price (or latency) changes, we have that $p \in \mathbb{R}^n$ is a random vector, with known mean $\bar{p}$ and covariance Σ on the assets (paths) in the portfolio. Therefore, with portfolio $z \in \mathbb{R}^n$, the return r is a (scalar) random variable with mean $\bar{p}^T z$ and variance $z^T \Sigma z$. The choice of portfolio z involves a trade-off between the mean of the return and its variance. The portfolio optimization problem is the following quadratic program:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & z^T \Sigma z \\ \text{subject to} \quad & \bar{p} z \geq r_{min} \\ & \mathbf{1}^T z = 1, \\ & z_i > 0 \quad \forall i = 1, \ldots, n. \end{aligned} \tag{1}$$

The risk of a small or large loss, *i.e.*, a change in portfolio values below its expected value, is directly related to the standard deviation, and increases with it. For this reason, the standard deviation (or the variance) is used as a measure of the risk associated with the portfolio.

Modeling Risky Paths in Mobile Edge Computing. We model the domain of financial instruments to be selected as physical paths, and the portfolio to be invested as virtual paths (or flows) connecting two end-points. We then model the expected return of our portfolio over a period of time (the time during which we hold the assets) as the throughput during the considered lifetime of a flow. The availability of all resources composing the portfolio (in our case physical links) fluctuates due to edge user mobility, failures, and the statistical multiplexing nature of connectionless networks. Packets corrupted or lost due to queuing delays or congestion increase throughput variance across each flow and the mobile and prone to failures nature of edge computing applications exacerbate such variations. For each path (flow) j, we model its risk (volatility) with z_{ij}, that is, as the probability of obtaining a given latency variance if we select z_{ij}. We now describe a method that we used to estimate such path latency, that can be in turn used as input of our optimization problem.

3 PREDICTING RISKY PATHS

Bayesian Networks. A Bayesian network is a probabilistic graphical model containing random variables and their conditional dependencies, expressed by a directed acyclic graph. The edge $x_j \rightarrow x_i$ represents the dependency of the random variable x_i on the random variable x_j, in which x_j is the parent of child x_i. Each random variable (a latency value) has a corresponding conditional probability table that is used to determine the conditional probability $\mathrm{P}(x_i|x_j)$, *i.e.*, the probability of x_i given x_j, where:

$$P(x_i|x_j) = \frac{P(x_i \cap x_j)}{P(x_j)}. \tag{2}$$

We define the set of parents for the random variable x_i to be $\mathrm{Pa}(x_i)$. To construct a Bayesian network from discretized data, we follow the constraint-based approach outlined by Koller and Friedman [?]. We omit such description in this paper for lack of space.

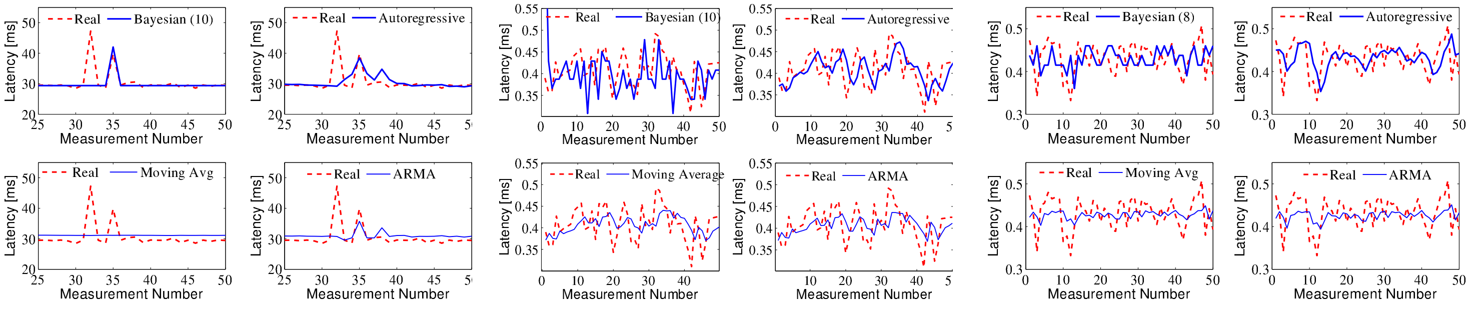
Our approach is based on three steps: (1) Discretize latency measurements using k-means clustering, (2) construct a Bayesian network with discretized latency data, and (3) predict latency using the Bayesian network by maximizing conditional probability $\mathrm{P}(x_t \mid \mathrm{Pa}(x_t))$. We use k-means clustering to discretize path latency measurements. The k-means clustering algorithm partitions observations $(y_1, y_2, \ldots, y_n)$ into sets $S = \{S_1, S_2, \ldots, S_k\}$ by minimizing the within-cluster sum of squares:

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{y \in S_i} ||y - \mu_i||^2, \tag{3}$$

where μ_i is the mean of the observations in S_i. We define the set of discrete latency values as $\{a_1, \ldots, a_n\}$, where n is the total number of clusters.

Constructing the Latency Bayesian Network. We construct a Bayesian network using the collected discretized latency measurements. Each node in the network represents the latency at a point in time relative to time t, with the aim of predicting the latency at time t. The set of nodes in the network is $\{x_{t-n}, x_{t-n+1}, \ldots, x_{t-1}, x_t\}$. The set of possible states for a node is composed by the means of the clusters.

Predicting Latency. Once the Bayesian network is constructed, we use it to predict path latency at time t, given the latency measurements for all nodes in the set $\mathrm{Pa}(x_t)$. Similar approaches have been used to predict daily stock price fluctuations [?]. We first obtain the discrete latency values for parent nodes of x_t, and then predict the discrete latency by computing the conditional probability $\mathrm{P}(x_t \mid \mathrm{Pa}(x_t))$ for each of the possible states of x_t. The predicted latency a_t is the state a_l of node x_t in which the conditional probability is maximized: $a_t = \mathrm{argmax}_{a_l} P(a_l|Pa(x_t))$, where each state a_l is

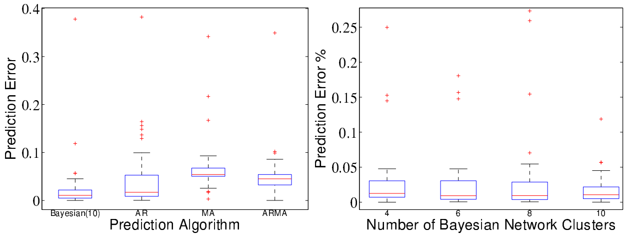


(a) SLU - Stanford

(b) GENI: GPO Rack

(c) GENI: GPO - University of Washington

Figure 1: The Bayesian Network approach predicts e2e latencies more accurately w.r.t. Autoregressive, Moving Average and Autoregressive moving average: (a) ICMP traffic from Saint Louis to Stanford.edu (b) ICMP traffic across VMs in the same GENI rack. (c) ICMP traffic across two East-coast GENI racks.



(a) Predictors Comparison

(b) Clusters Impact

Figure 2: (a) The Bayesian network prediction with 10 clusters has the lowest latency prediction error w.r.t. Autoregressive, Moving Average and Autoregressive moving average. (b) Impact of the number of clusters in the Bayesian network on the prediction error.

the mean of cluster S_l. To construct a portfolio, we predict the next value of latency for each available path in a network. We then select the set of paths (portfolio) that minimize the overall predicted latency (the risk).

4 EVALUATION

We evaluate our method against an autoregressive model, a moving average model, and an ARMA (autoregressive moving average) model. These methods captures the time series of latency values. Unlike Bayesian networks, they do not model the dependencies within a dataset. Forecasts are made by these time series models by combining past values. Consistent with the Bayesian network method, each predicted latency value is a one-step-ahead forecast. Although an extensive evaluation campaign would be needed to draw more meaningful conclusions, our initial results suggest that the Bayesian network approach predicts e2e latencies more accurately w.r.t. autoregressive, moving average and autoregressive moving average models (Figure **??**). Our measurements include ICMP traffic (*i.e.* ping) from Saint Louis to the `Stanford.edu` web server, and within the GENI testbed [**?**]; we emulated an edge network pinging VMs in the same GENI rack as well as across two close enough East-coast GENI racks. To quantify the superior behavior of the Bayesian network approach, we show the distribution of errors across the tested methods in Figure **??***a*. Finding the optimal number of clusters that minimize the prediction error is still an open question, but our initial results suggest that, as long as there is dependency among latency measurements, 8 or 10 clusters produce the best accuracy, especially for latency peaks (Figure **??***b*).

5 LIMITATIONS, LESSON LEARNED AND FUTURE WORK

Our finding are far from ideal for several reasons. The accuracy of the Bayesian network is limited by its ability to correctly identify and model dependencies in the data. If the variance within a dataset is small, or the dependencies between variables are weak, the Bayesian network is likely to miss the dependencies. Additionally, the clustering process is likely to hide small dependencies because the variance between two data points can be lost when they belong to the same cluster, as we use the cluster mean as input to the Bayesian network. Our results were also limited by the consistency of the latency values in our data. We found that this resulted in the cluster means having a small spread. When the Bayesian network predicted the incorrect cluster, the value of the predicted cluster was close to the actual latency value, resulting in low errors despite the erroneous prediction.

In future work, we plan to explore improvements to our Bayesian network prediction method. We also plan to continue investigating the dependencies in latency data through alternative methods. We also intend to expand our latency prediction method to a multipath network in order to implement a more complete path management solution.