Bringing Data Analytics to the Network Nodes

Alba P. Vela*, Anna Via, Marc Ruiz, and Luis Velasco

Universitat Politècnica de Catalunya (UPC), Barcelona (Spain), Email: apvela@ac.upc.edu

Abstract *Monitoring every 15 minutes imposes long traffic anomaly detection times and thus, the monitoring frequency needs to be increased to reduce those times, which entails large amount of monitoring data collected. Consequently, we propose bringing data analytics to the nodes.*

Introduction

Traffic monitoring is an essential task for network operators since it allows to evaluate network performance. Monitoring traffic samples can be collected in a repository for further analysis[1], e.g. to create predicted traffic matrices for the near future. Among the different use cases of data analytics for networking, that of identifying problems (or anomalies) is undoubtedly of the interest of many network operators. Traffic anomalies are *short-living* events that do not follow expected patterns (see a survey in [2]). Detecting anomalies is a difficult task because anomalous patterns need to be extracted and interpreted from large amounts of high-dimensional, noisy data.

According to the ITU-T[3], performance events are counted second by second over every 15-minute period. At the end of a period, they are stored in the historical registers, usually in the network management plane. It is clear that when analytics are applied to data collected every 15 minutes, expected traffic anomaly detection times will be as well in that order of magnitude. Note that traffic anomalies can create network congestion and stress resource utilization in routers and hence, its prompt detection becomes essential since it allows preparing the network e.g., by modifying routing tables or reconfiguring the virtual network topology[4].

In this paper, we evaluate the performance of different Origin-Destination (OD) traffic anomaly detection methods and propose monitoring strategies and architectural approaches to find the combination with the best trade-off between detection time and amount of collected data.

Architecture for Traffic Modeling and Anomaly Detection

In our approach, OD traffic models are computed for the expected average, which predict two response variables: the mean ($\mu(t)$) and the standard deviation ($\sigma(t)$). Fig. 1 illustrates the main steps of the proposed method. In Fig. 1a, traffic samples for every OD pair are collected from the packet nodes at a given monitoring period. Besides, Fig. 1b shows upper and lower bounds computed as $\mu \pm 3\sigma$ and traffic samples for a given OD pair and for a typical day. Although out-of-bound samples are considered as *atypical* (Fig. 1c), its detection does not entail a traffic anomaly. In fact, the decision of whether an atypical sample is considered as a traffic anomaly cannot be based on just one single sample, but in observing some previous samples. This is depicted in Fig. 1d, where an anomaly is detected after receiving two out-of-bound samples and considering some other previous within-bound samples.

To efficiently implement OD-based traffic anomaly detection methods, we propose the modules depicted in Fig. 2 that are first assumed to be placed in the management plane. In such architecture, traffic samples are collected from the packet nodes and stored in the *collected data repository*. Collected data can be conveniently summarized in modeled data (e.g. by computing average values). The *Estimator* module applies data analytics on samples from the modeled data repository to estimate the specific models for every OD pair, which are stored in a *model repository*. Models predict response variables for the average OD traffic (i.e. $\mu(t)$ and $\sigma(t)$). The *Sentinel* module is in charge of detecting traffic anomalies; it first verifies whether a just arrived OD traffic sample is out of bounds and only in such case, a machine learning algorithm is run to detect traffic anomalies.

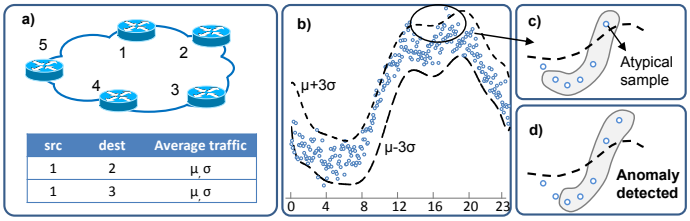


Fig. 1. (a) Per OD statistical models. (b) Monitoring samples vs. estimation. (c) Atypical and (d) anomaly detection.

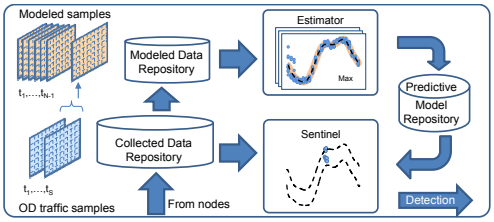


Fig. 2. Architecture for OD traffic anomaly detection.

ISBN 978-3-8007-4274-5

Methods for Anomaly Detection

Two different methods for anomaly detection are studied in this section: *Threshold-based* and *Probability-based*. Both methods have already been proposed in different contexts as in traffic changes detection[1] and anomaly detection based on hypothesis testing[5]; in this paper we adapt them for OD traffic anomaly detection. The adapted *Threshold-based* method consists in detecting anomalies after receiving a number of out-of-bound traffic samples with respect to the $\mu\pm3\sigma$ confidence interval. The *Probability-based* method is a self-learning classifier with two labels for the response: normal and anomaly. The algorithm (see Fig. 3) is based on a multi-response model to predict whether a sequence of consecutive traffic samples belongs to the normal class or, on the contrary, there is sufficient evidence to declare it as anomalous.

The algorithm starts when a traffic sample $x(t)$ is received in time t; let $x'(t)$ be the normalized value of $x(t)$ with respect to the average model i.e. $x'(t)=(x(t)-\mu(t))/\sigma(t)$. Note that a normalized value equal to k means that that $x(t) = \mu(t)+k\cdot\sigma(t)$. After normalization, $x'(t)$ is stored in a fixed-size data series H, containing the last normalized traffic samples received (hereafter referred to as *features*). As a result of traffic normalization, every feature H_i follows the standard Gaussian distribution, i.e. $H_i \sim \mathcal{N}(0,1)$. Let us define a random variable $Z\sim\mathcal{N}(0,1)$ and the probability $p_i=1-P(Z\leq|H_i|)$ that feature i takes a value above its current absolute value i.e. feature i actually belongs to the normal class. Therefore, it is likely to assume that smaller (i.e. less probable) p_i values will be observed in case of an anomaly. The classifier basically consists of: *i*) a distance function $w(H)$ to compute how likely is that a features vector H belongs to the normal class; and *ii*) a distance threshold w_{thr} that normal data series H do not practically exceed. The distance function $w(H)$ is defined as the product of p_i ($w(H)=\Pi_{1..|H|}\, p_i$).

To decide whether an anomaly is detected, we simply compare $w(H)$ against the w_{thr} threshold i.e. when $w(H)<w_{thr}$. Under the assumption of feature independence and considering p_i=0.05 for each individual feature as a common used limit for accepting the Gaussian null hypothesis, we fix $w_{thr} = 0.05^{|H|}$.

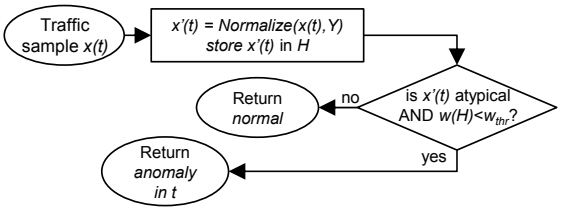


Fig. 3 Probability-based algorithm for anomaly detection.

Proposed monitoring strategies

As introduced, anomaly detection time is directly related with the monitoring frequency and thus, it is key parameter for study. However, increasing monitoring frequency entails increasing in the same proportion the amount of monitoring data to be conveyed to the repositories in management plane. Aiming at keeping the latter amount under control, in this paper we propose studying the performance of the following monitoring strategies: *i*) the tradicional *fixed monitoring* period strategy but reducing its period to accelerate anomaly detection; *ii*) a *dynamic monitoring* strategy, where the monitoring period can be re-programmed during the day; and *iii*) a *reactive monitoring* strategy (*c*:*f*) that uses a coarse monitoring period (*c*) and re-configures it to a finer period (*f*) after detecting the first out-of-bound traffic sample.

From the possible combination of methods and strategies, we focus on studying the four most relevant approaches: *i*) Threshold-based with fixed monitoring (*Threshold-fixed*), *ii*) Probability-based with fixed monitoring (*Probability-fixed*), *iii*) Probability-based with dynamic monitoring (*Dynamic*), and *iv*) Probability-based with reactive monitoring (*Reactive*).

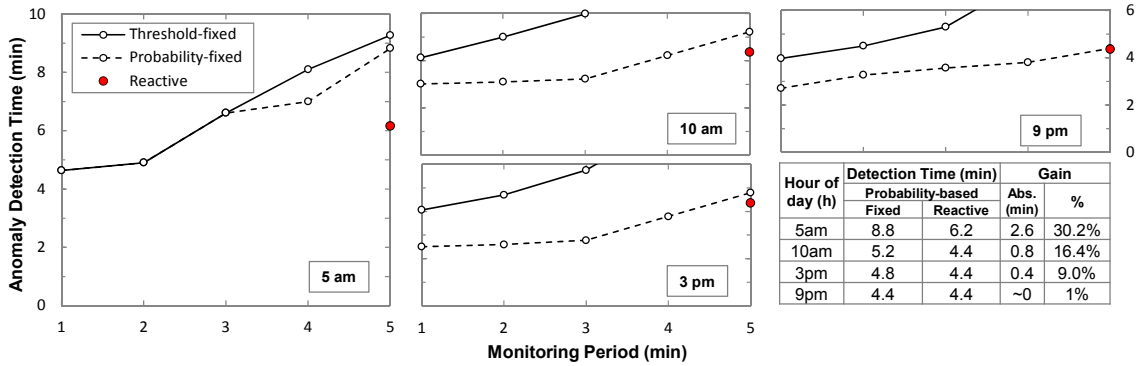
Illustrative numerical results

For evaluation purposes, we developed an ad-hoc event-driven simulator in OMNET++ that was used to generate normal traffic and traffic anomalies separately. Traffic is generated as the summation of two different functions, mean and noise, where the traffic mean represents a normal day with values varying along day hours, while the noise is a random function with mean zero and a given standard deviation.

Regarding anomalies, they are generated following a pulse function where the raising front consists of an exponential function and are used as a multiplicative factor over traffic. Anomalies can be configured to be triggered at any specific time and with any specific duration and scaling factor. As an illustrative example, an anomaly can be generated to double mean traffic, last for 2 hours and characterized by reaching the 90% of its maximum value in the first 30 minutes.

The Estimator module was implemented in C++ and integrated in the simulator, whereas the Sentinel module implementing the proposed approaches, was developed in R and kept as a separated standalone module.

Graphs in Fig. 4 plot, for several hours of the day, the anomaly detection time for the Threshold-fixed, Probability- fixed and Reactive approaches, where the monitoring period is in the interval [1-5] minutes. We observe that although anomaly detection time varies for the different considered hours of day,



Hour of day (h)	Detection Time (min)		Gain	
	Probability-based		Abs. (min)	%
	Fixed	Reactive		
5am	8.8	6.2	2.6	30.2%
10am	5.2	4.4	0.8	16.4%
3pm	4.8	4.4	0.4	9.0%
9pm	4.4	4.4	~0	1%

Fig. 4 Monitoring period as a function of the target traffic anomaly detection time for different hours of a day.

detection time increases remarkably with the Threshold-fixed approach when the monitoring period increases. This is in contrast to the moderated increment achieved by the Probability-fixed one. In the case of the Reactive approach, where we assume a (*c*=5:*f*=1) min. monitoring strategy, slightly lower detection times w.r.t. the previous approaches can be observed. The table in Fig. 4 reports the gains in detection time for the studied hours of day, where using a finer monitoring period after an out-of-bound traffic sample is detected provides gains between 1% and 30%, depending on the hour of day.

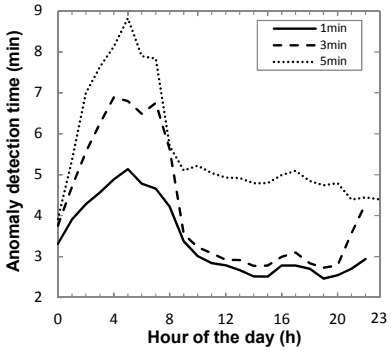


Fig. 5 Anomaly detection time vs. hours of day for different monitoring periods.

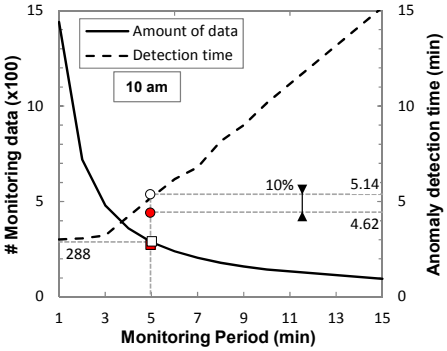


Fig. 6 Amount of data and anomaly detection time vs. monitoring period.

Fig. 5 focuses on studying in depth the potentials of the Probability-method and illustrates how anomaly detection depends on different factors such as the changes on volume traffic among the different hours of the day for the same monitoring period. This opens the opportunity to dynamically adapt the monitoring period for different hours of day and achieve the same anomaly detection times (Dynamic monitoring). Note that this is positive since to achieve low detection times, 1 minute period should be fixed. Hence, by relaxing the monitoring period we are effectively reducing the amount of monitoring data to be collected.

Finally, Fig. 6 shows the amount of monitored data to be collected along the day when reducing the monitoring period. E.g., assuming a 5 min. period, 288 monitoring samples per OD and day need to be collected achieving 5.14 and 4.62 min. detection times for the Probability-fixed and the Reactive approaches, respectively.

Concluding discussion

The above showed that 15-minute monitoring cannot provide the short anomaly detection times required to react against unexpected traffic changes. Consequently, we studied four different approaches mixing detection methods and monitoring strategies and showed that the shortest anomaly detection times are achieved when monitoring every 1 min. Nevertheless, this comes at the cost of collecting and storing large amount of data in management plane.

In view of the above, we propose to bring the proposed data analytics method for OD anomaly detection to the network nodes thus, relaxing data collection from the management plane to the traditional 15-min. period, that can be used for traffic modelling and estimation purposes. Table 1 compares function placement in the centralized and distributed architectures.

Table 1. Comparative function placement

	Centralized	Distributed
Mgmt Plane	• Monitoring freq. program. • Traffic estimation. • OD Anomaly detection.	Traffic estimation.
Node	Monitoring frequency reconfiguration.	OD Anomaly detection.

Acknowledgements

The research leading to these results has received funding from the Spanish MINECO SYNERGY project (TEC2014-59995-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

References

1 Ll. Gifre et al., “Big Data Analytics in Support of Virtual Network Topology Adaptability,” OFC 2016.
2 V. Chandola et al. “Anomaly Detection: A Survey,” ACM Computing Surveys, vol. 41, pp. 1-72, 2009.
3 ITU-T Rec. M.2120, 2002.
4 F. Morales et al., “Virtual Network Topology Reconfiguration based on Big Data Analytics for Traffic Prediction,” OFC 2016.
5 R. Kuehl, “Design of Experiments”, Thomson Learning, 2000.